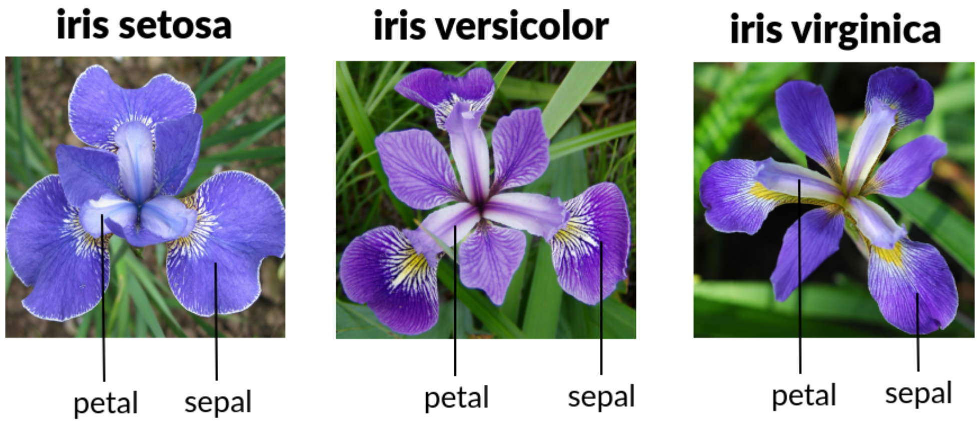
iris_eda

March 30, 2024

[1]:

```python
from IPython.display import Image
Image(filename='Iris-class.png')
```

[1]:



1 Niezbędne Biblioteki

[2]:

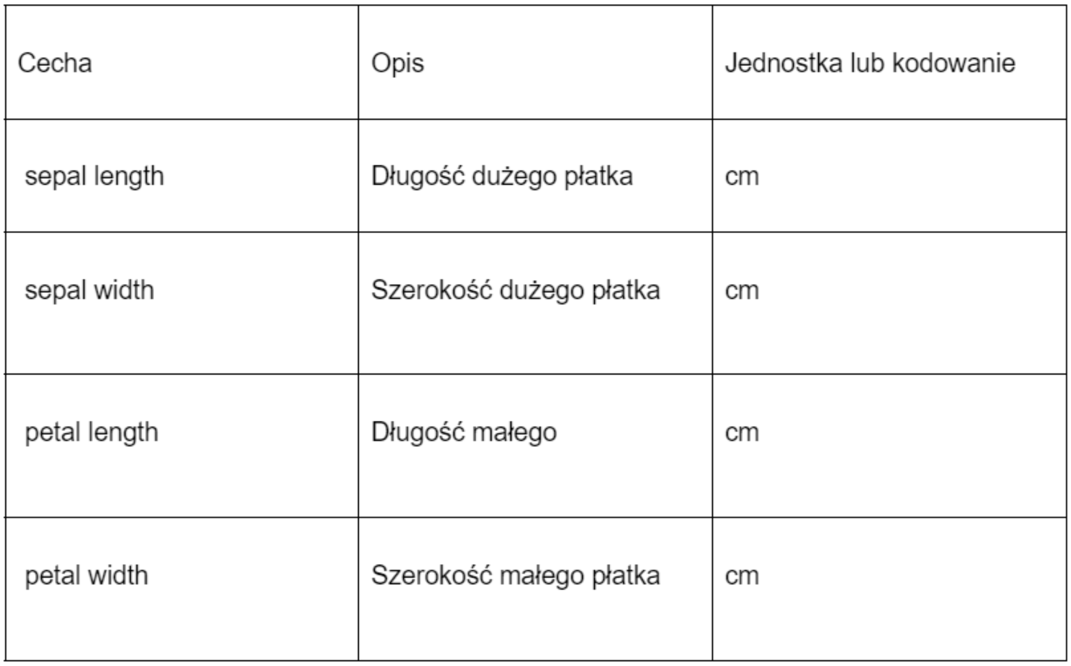
```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from ucimlrepo import fetch_ucirepo
```

2 Tabela Opisowa

[3]:

```python
Image(filename='tabela_opisowa.png')
```

[3]:



Cecha	Opis	Jednostka lub kodowanie
sepal length	Długość dużego płatka	cm
sepal width	Szerokość dużego płatka	cm
petal length	Długość małego	cm
petal width	Szerokość małego płatka	cm

3 Wczytanie Danych

[4]:
```
iris = fetch_ucirepo(id=53)
iris_df = pd.concat([iris.data.features, iris.data.targets], axis=1)
```

4 Wstępne zapoznanie się z danymi

[5]:
```
print(iris_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal length  150 non-null    float64
 1   sepal width   150 non-null    float64
 2   petal length  150 non-null    float64
 3   petal width   150 non-null    float64
 4   class         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

[6]:

```
print(iris_df.shape)
```

```
(150, 5)
```

[7]:

```
print(iris_df.columns)
```

```
Index(['sepal length', 'sepal width', 'petal length', 'petal width', 'class'],
dtype='object')
```

[8]:

```
print(iris_df.head())
```

```
   sepal length  sepal width  petal length  petal width        class
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa
```

[9]:

```
print(iris_df.describe())
```

```
       sepal length  sepal width  petal length  petal width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

[10]:

```
print(iris_df['class'].value_counts())
```

```
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

[11]:

```
print(iris_df.isnull().sum())
```

```
sepal length    0
sepal width     0
petal length    0
petal width     0
class           0
dtype: int64
```

5 Statystyki opisowe dla każdego gatunku

[12]:

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
print(iris_df.groupby('class').describe())
```

```
                sepal length                                              \
                       count   mean       std  min    25%  50%  75%  max   
class
Iris-setosa             50.0  5.006  0.352490  4.3  4.800  5.0  5.2  5.8
Iris-versicolor         50.0  5.936  0.516171  4.9  5.600  5.9  6.3  7.0
Iris-virginica          50.0  6.588  0.635880  4.9  6.225  6.5  6.9  7.9

                sepal width                                                \
                      count   mean       std  min    25%  50%    75%  max   
class
Iris-setosa            50.0  3.418  0.381024  2.3  3.125  3.4  3.675  4.4
Iris-versicolor        50.0  2.770  0.313798  2.0  2.525  2.8  3.000  3.4
Iris-virginica         50.0  2.974  0.322497  2.2  2.800  3.0  3.175  3.8

                petal length                                               \
                       count   mean       std  min  25%   50%    75%  max   
class
Iris-setosa             50.0  1.464  0.173511  1.0  1.4  1.50  1.575  1.9
Iris-versicolor         50.0  4.260  0.469911  3.0  4.0  4.35  4.600  5.1
Iris-virginica          50.0  5.552  0.551895  4.5  5.1  5.55  5.875  6.9

                petal width
                      count   mean       std  min  25%  50%  75%  max
class
Iris-setosa            50.0  0.244  0.107210  0.1  0.2  0.2  0.3  0.6
Iris-versicolor        50.0  1.326  0.197753  1.0  1.2  1.3  1.5  1.8
Iris-virginica         50.0  2.026  0.274650  1.4  1.8  2.0  2.3  2.5
```

[13]:

```
print(iris_df.groupby('class').median())
```

```
                 sepal length  sepal width  petal length  petal width
class
Iris-setosa               5.0          3.4          1.50          0.2
Iris-versicolor           5.9          2.8          4.35          1.3
Iris-virginica            6.5          3.0          5.55          2.0
```

5.1 90-ty percentyl dla każdej z cech dla każdego gatunku

[14]:

```
print(iris_df.groupby('class').quantile(0.9))
```

```
                 sepal length  sepal width  petal length  petal width
class
Iris-setosa              5.41         3.90          1.70         0.40
```

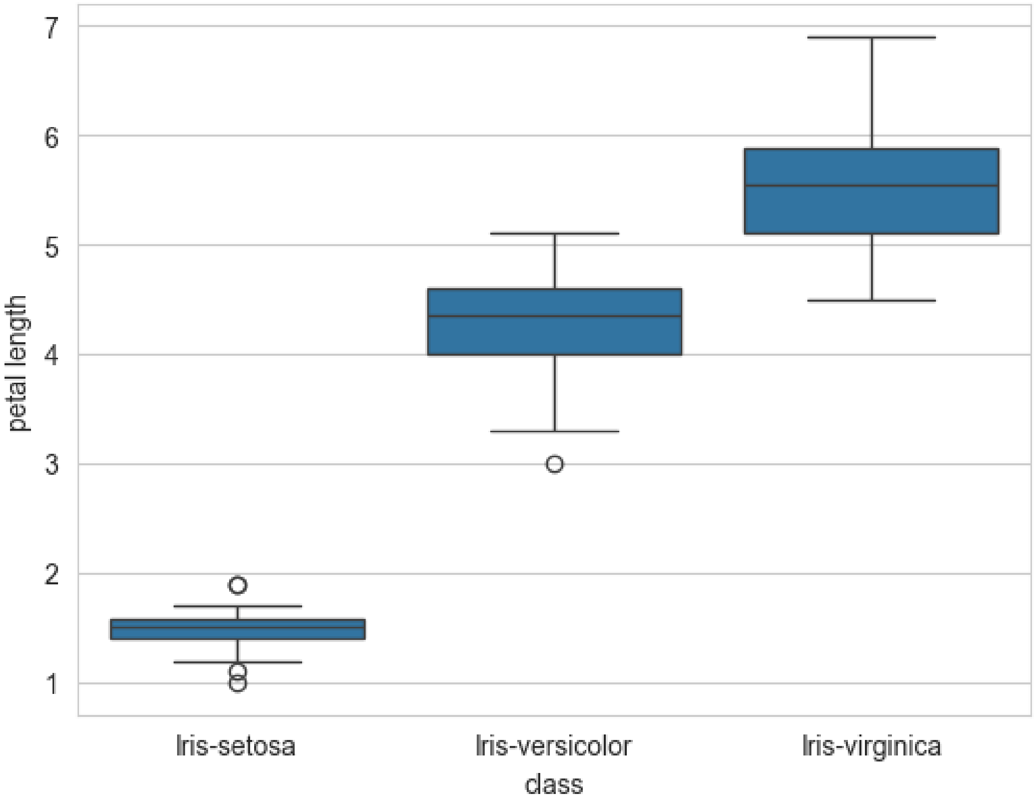
```
Iris-versicolor          6.70         3.11          4.80         1.51
Iris-virginica           7.61         3.31          6.31         2.40
```

6 Wizualizacja Danych

6.1 Histogramy - rozkład cech

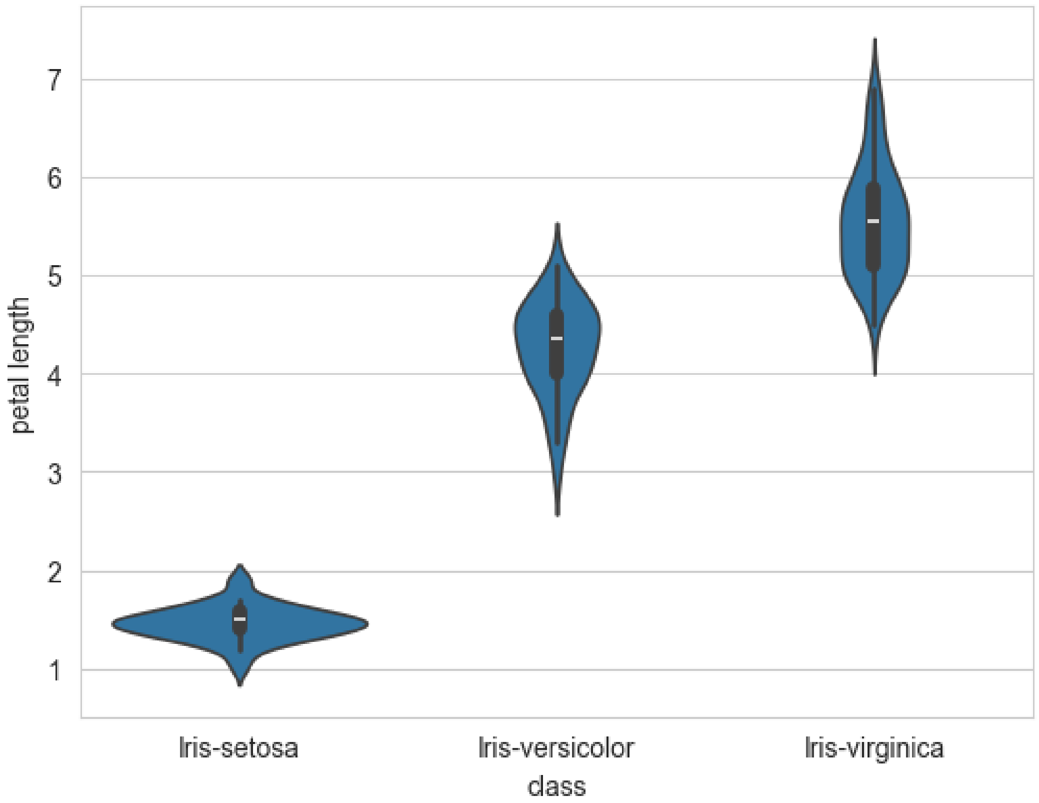
[15]:
```
sns.boxplot(x='class', y='petal length', data=iris_df)
```

[15]:
```
<Axes: xlabel='class', ylabel='petal length'>
```



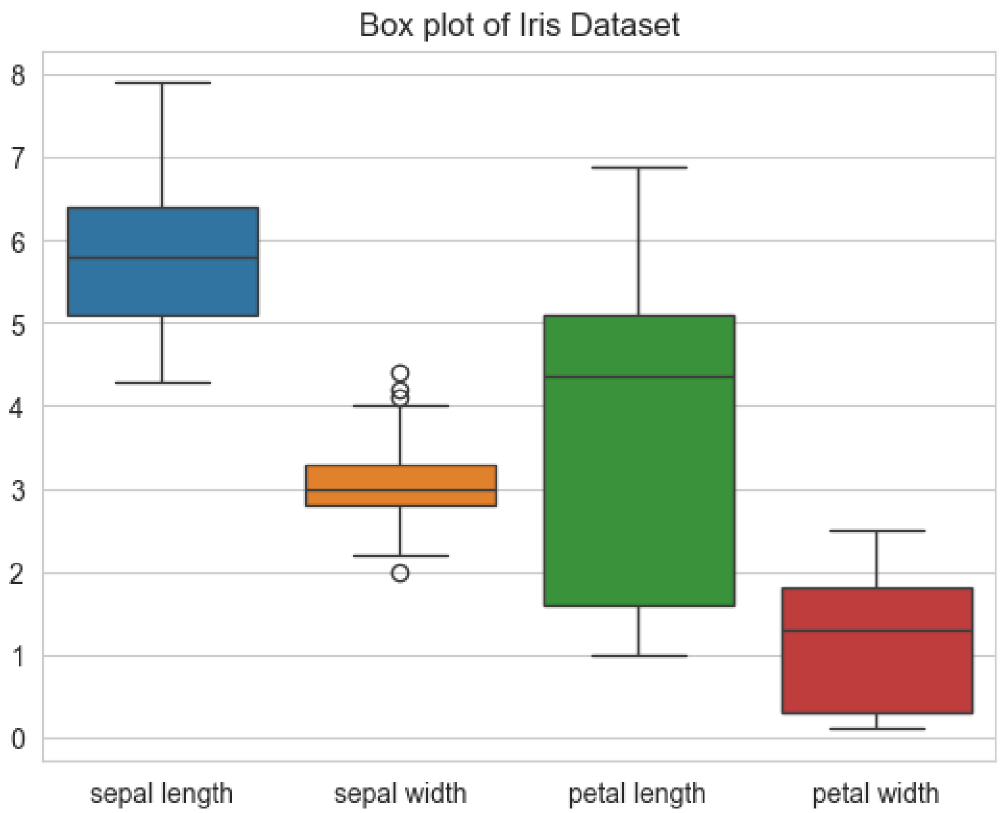
[16]:
```
sns.violinplot(x="class", y="petal length", data=iris_df)
```

[16]:
```
<Axes: xlabel='class', ylabel='petal length'>
```



```
[17]: sns.boxplot(data=iris_df.drop('class', axis=1))
      plt.title('Box plot of Iris Dataset')
```

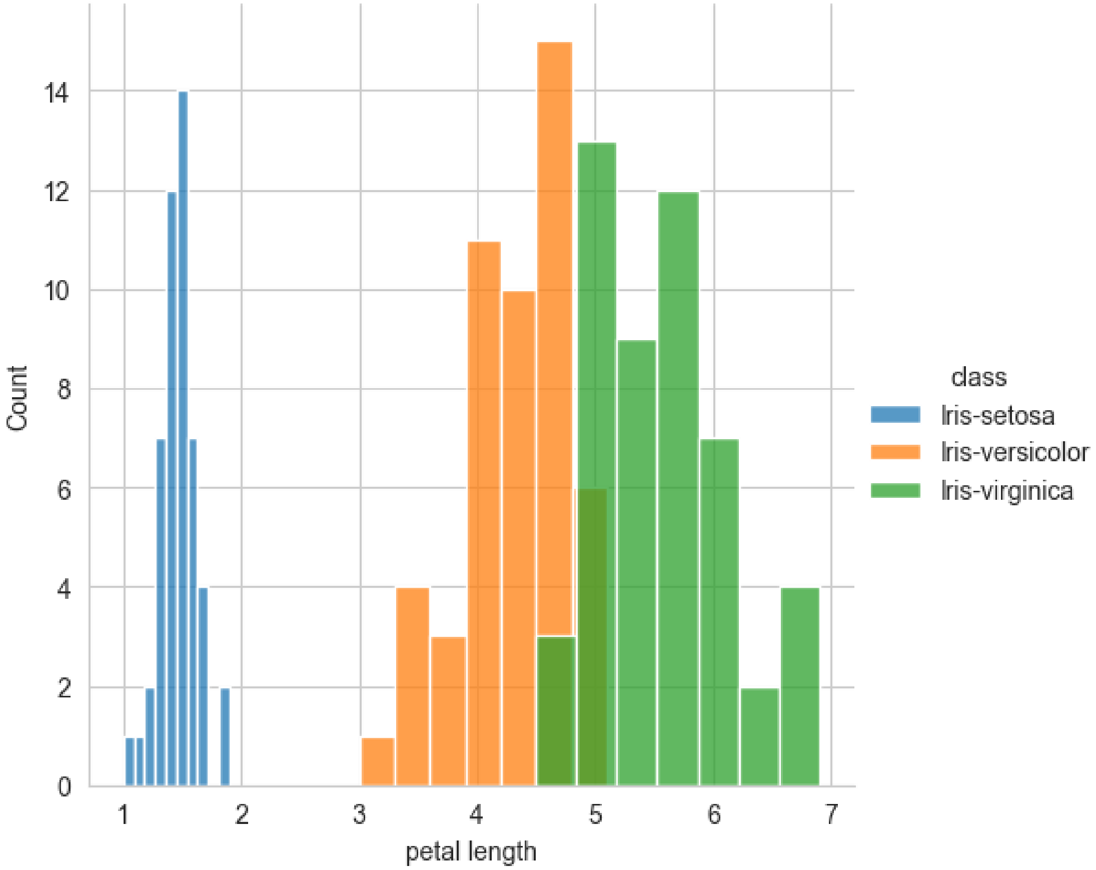
```
[17]: Text(0.5, 1.0, 'Box plot of Iris Dataset')
```



6.2 Histogramy z podziałem na odmiany

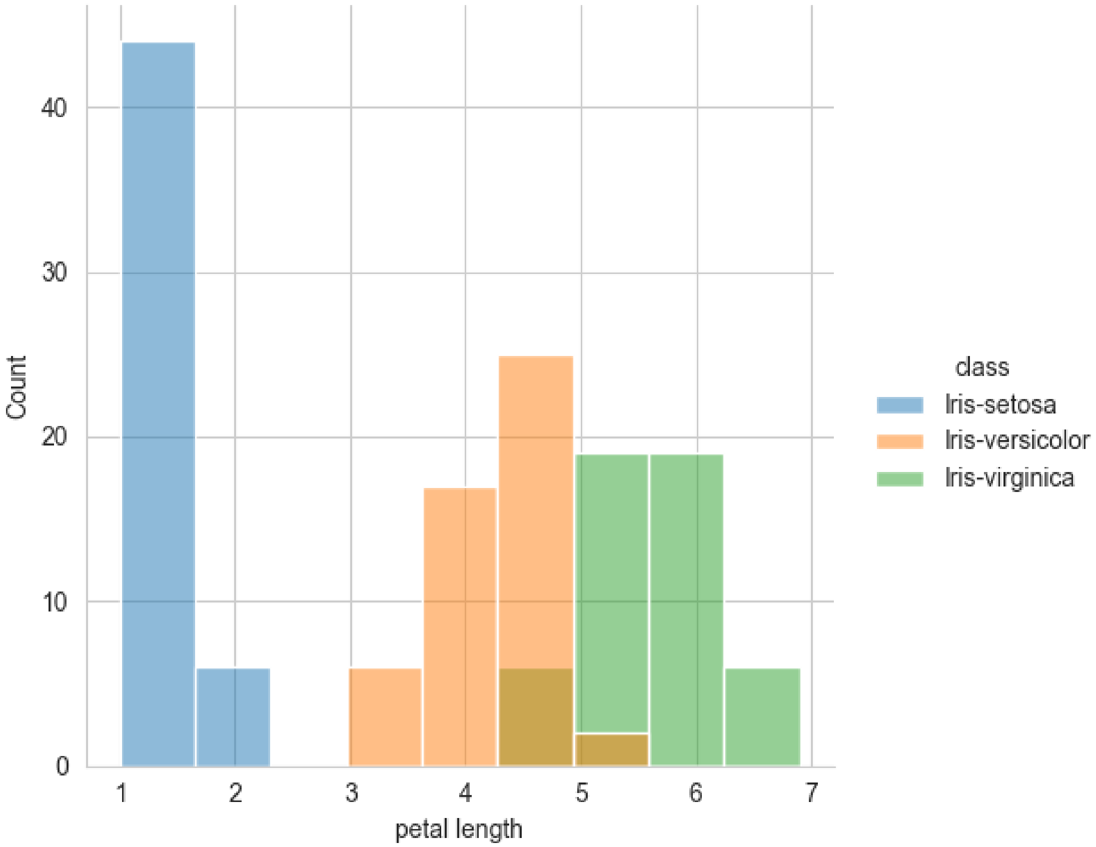
```
[18]: sns.FacetGrid(iris_df, hue="class", height=5).map(sns.histplot, "petal length").
      ↪add_legend()
```

```
[18]: <seaborn.axisgrid.FacetGrid at 0x2536a173b60>
```



[19]:
```
sns.displot(data=iris_df, x='petal length', hue='class', kind='hist', height=5)
```

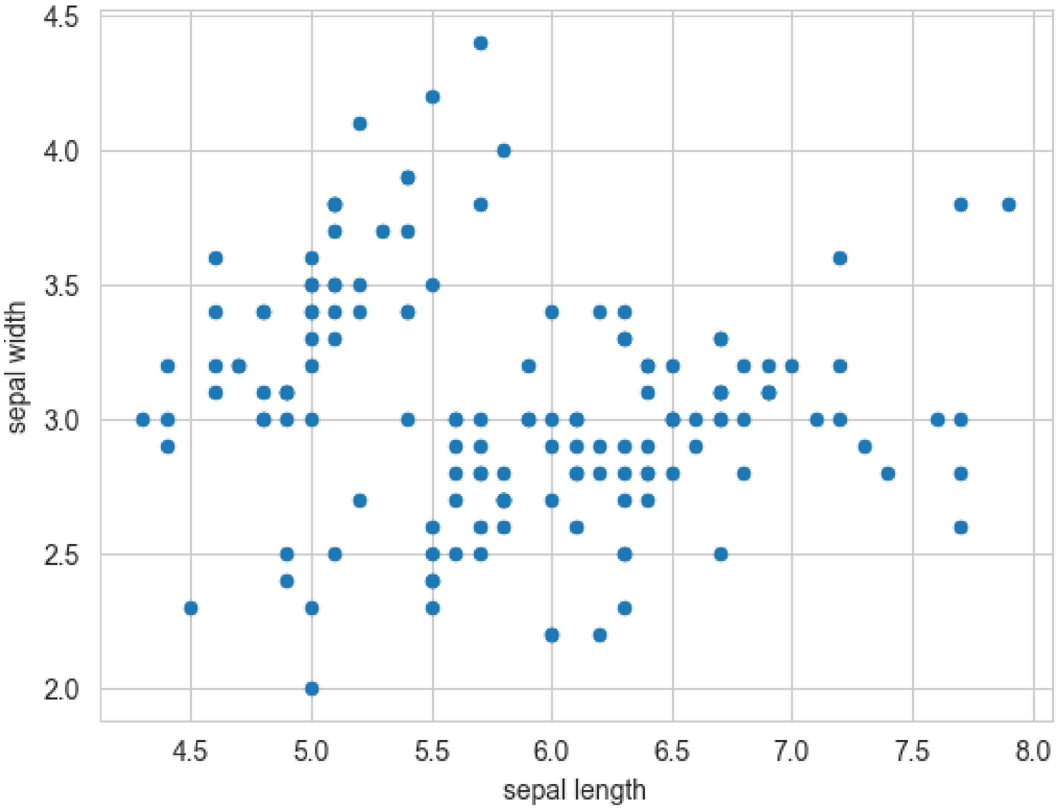
[19]:
```
<seaborn.axisgrid.FacetGrid at 0x2536a0873b0>
```



6.3 Przykład słabej wizualizacji danych

[20]:
```
iris_df.plot(kind='scatter', x='sepal length', y='sepal width')
```

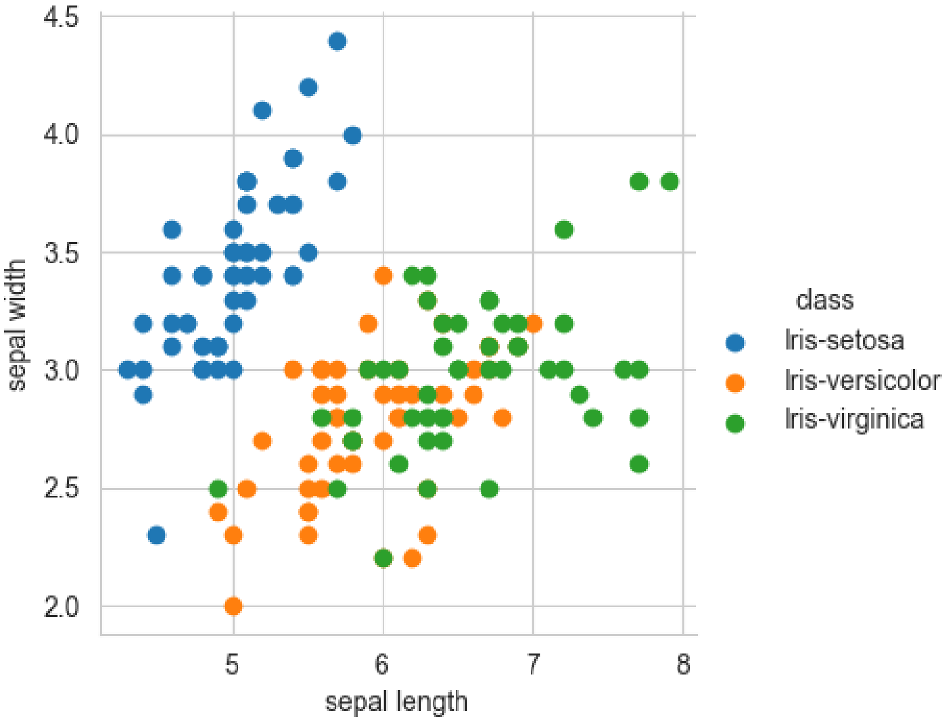
[20]:
```
<Axes: xlabel='sepal length', ylabel='sepal width'>
```



6.4 Przykład lepszej wizualizacji danych

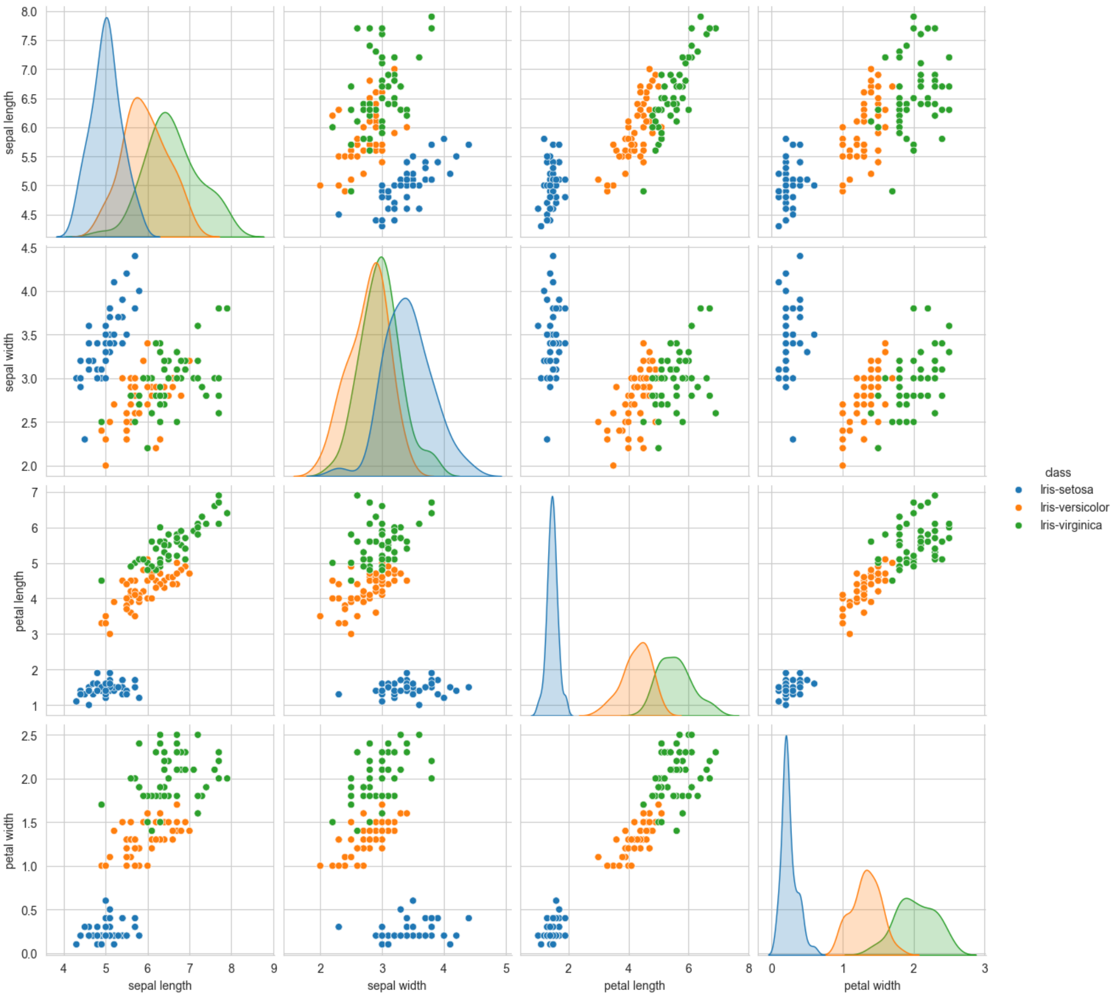
[21]:
```
sns.set_style('whitegrid')
sns.FacetGrid(iris_df, hue='class', height=4).map(plt.scatter, 'sepal length',␣
  ↪'sepal width').add_legend()
```

[21]:
```
<seaborn.axisgrid.FacetGrid at 0x2536e3afe90>
```



[22]:
```
sns.pairplot(iris_df, hue='class', height=3)
```

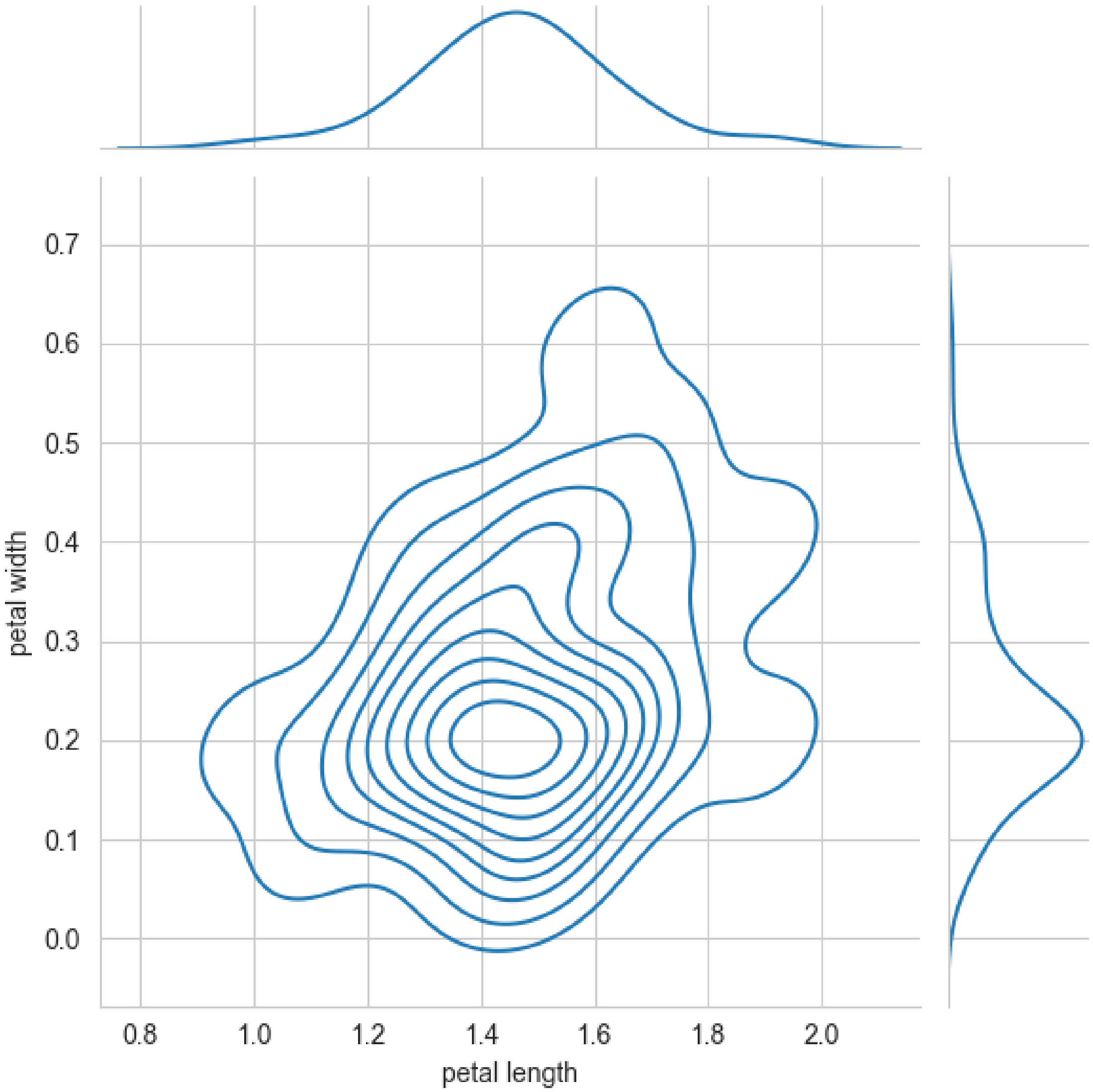
[22]:
```
<seaborn.axisgrid.PairGrid at 0x25358e53aa0>
```



6.5 Bardziej zaawanosowana wizualizacja danych

[23]:
```
sns.jointplot(x='petal length', y='petal width', data=iris_df[iris_df['class']␣
 ↪== 'Iris-setosa'], kind='kde')
```

[23]:
```
<seaborn.axisgrid.JointGrid at 0x2536e4659a0>
```



6.6 PDF AND CDF

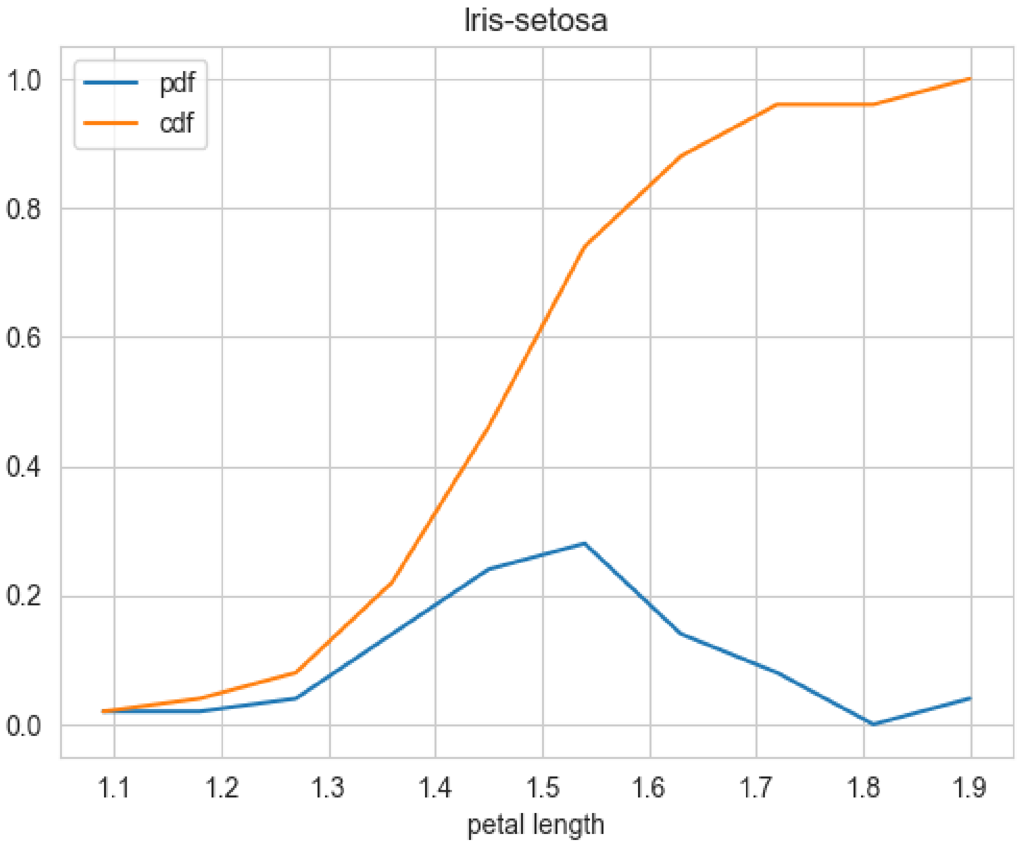
[24]:

```
iris_setosa = iris_df[iris_df['class'] == 'Iris-setosa']
counts, bin_edges = np.histogram(iris_setosa['petal length'], bins=10,␣
  ↪density=True)

pdf = counts / sum(counts)
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:], pdf, label='pdf')
plt.plot(bin_edges[1:], cdf, label='cdf')
plt.xlabel('petal length')
plt.legend(loc='best')
plt.title('Iris-setosa')
```

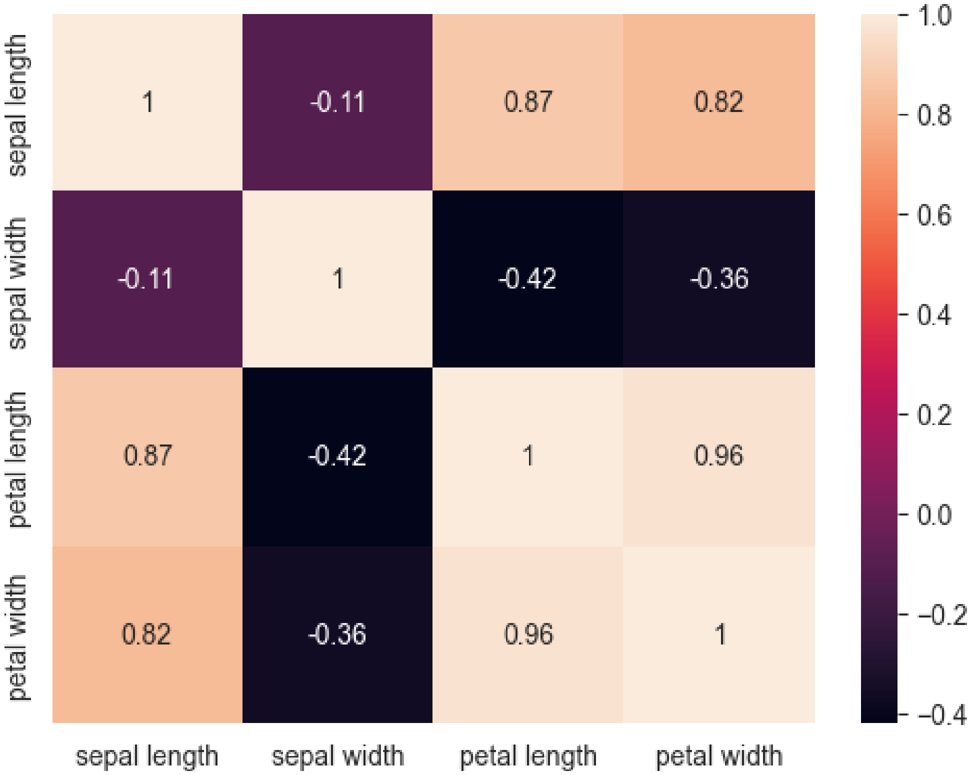
```
[24]: Text(0.5, 1.0, 'Iris-setosa')
```



6.7 Korelacja - analiza korelacji

```
[25]: corr_matrix = iris_df[['sepal length', 'sepal width', 'petal length', 'petal␣
      ↪width']].corr()
      sns.heatmap(corr_matrix, annot=True)
```

```
[25]: <Axes: >
```



	sepal length	sepal width	petal length	petal width
sepal length	1	-0.11	0.87	0.82
sepal width	-0.11	1	-0.42	-0.36
petal length	0.87	-0.42	1	0.96
petal width	0.82	-0.36	0.96	1

6.8 Dimensionality Reduction

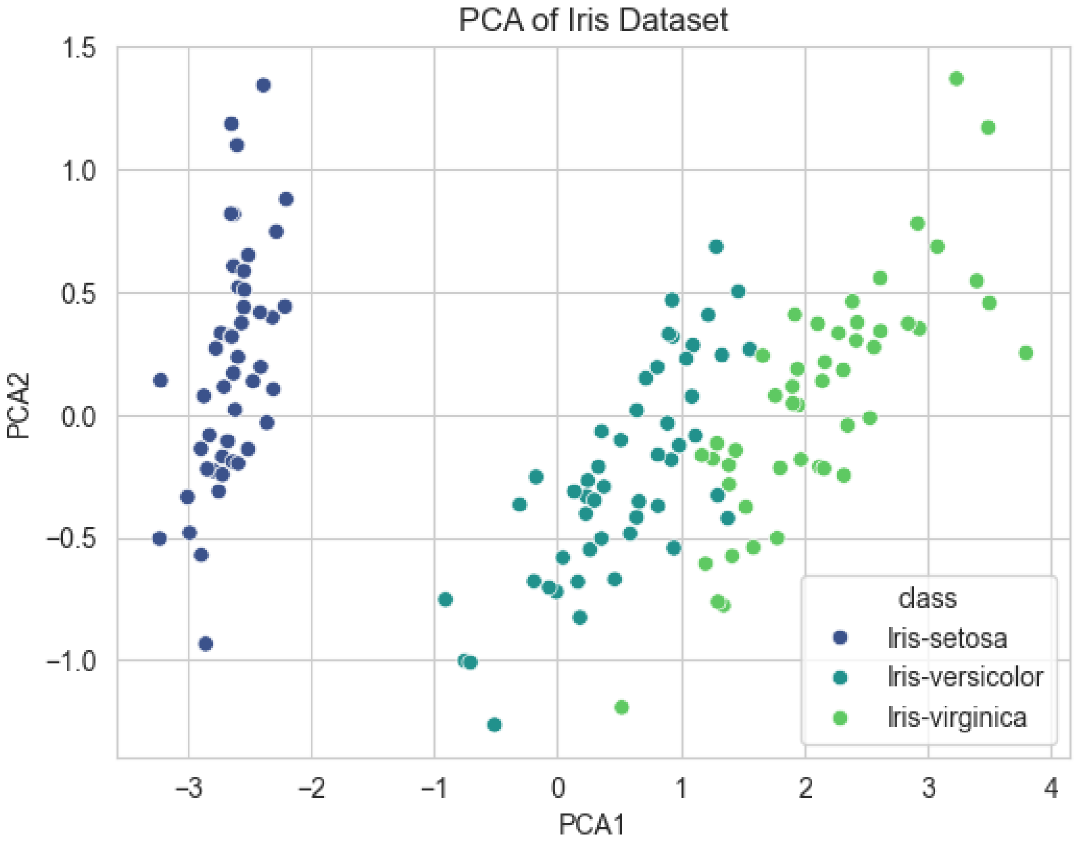
6.9 Principal Component Analysis (PCA)

[26]:
```
from sklearn.decomposition import PCA
iris_numeric = iris_df.drop('class', axis=1)

pca = PCA(n_components=2)
iris_pca = pca.fit_transform(iris_numeric)

iris_pca_df = pd.DataFrame(data=iris_pca, columns=['PCA1', 'PCA2'])
iris_pca_df['class'] = iris_df['class']

sns.scatterplot(data=iris_pca_df, x='PCA1', y='PCA2', hue='class',␣
  ↪palette='viridis')
plt.title('PCA of Iris Dataset')
plt.show()
```



6.10 t-Distributed Stochastic Neighbor Embedding (t-SNE)

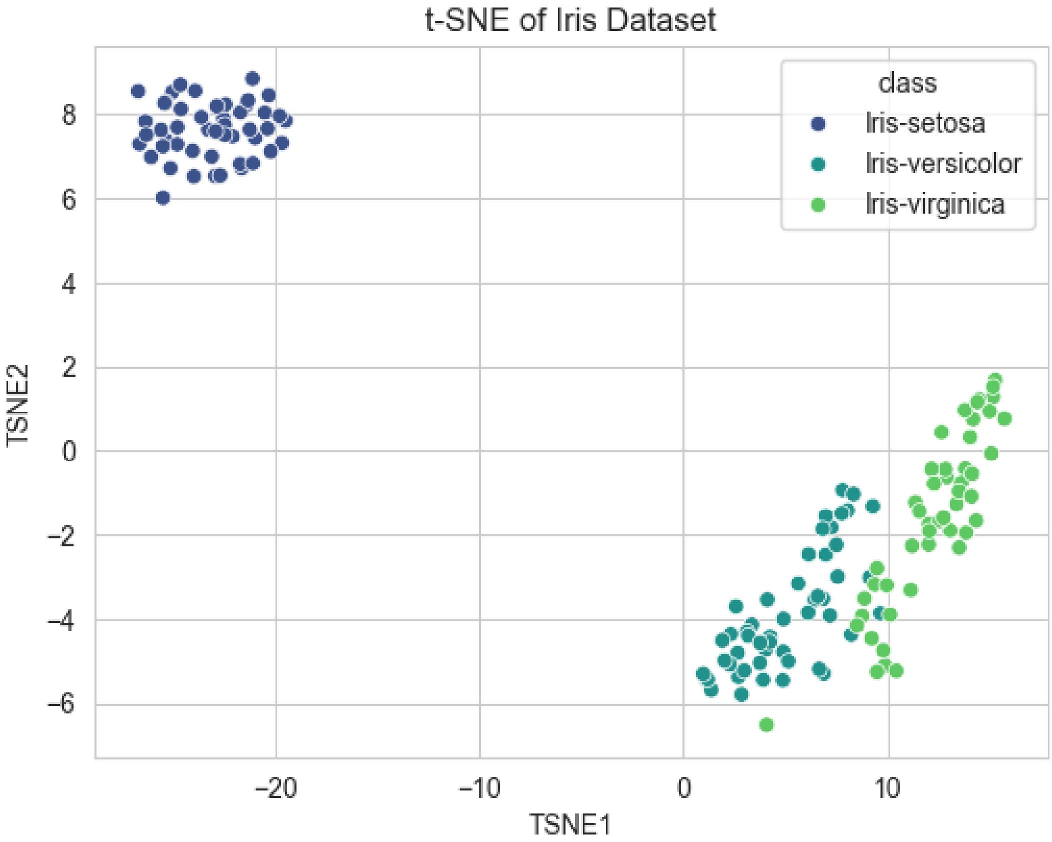
[27]:
```
from sklearn.manifold import TSNE


iris_features = iris_df.drop(columns=['class'])

tsne = TSNE(n_components=2, random_state=42)
iris_tsne = tsne.fit_transform(iris_features)

iris_df['TSNE1'] = iris_tsne[:, 0]
iris_df['TSNE2'] = iris_tsne[:, 1]

sns.scatterplot(x='TSNE1', y='TSNE2', hue='class', data=iris_df,␣
 ↪palette='viridis')
plt.title('t-SNE of Iris Dataset')
plt.show()
```



7 AUTORZY

7.1 Dominik Żebrowski

7.2 Maciej Rózio

[27]:

```
```