

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
df = pd.read_csv('/content/drive/MyDrive/MLDS-LAB/Iris.csv')
```

```
df.head()
```

```
{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\n    {\n      \"column\": \"Id\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 43,\n        \"min\": 1,\n        \"max\": 150,\n        \"num_unique_values\": 150,\n        \"samples\": [\n          74,\n          19,\n          119\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"SepalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.828066127977863,\n        \"min\": 4.3,\n        \"max\": 7.9,\n        \"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n          4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"SepalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.4335943113621737,\n        \"min\": 2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n        \"samples\": [\n          2.3,\n          4.0,\n          3.5\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"PetalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1.7644204199522626,\n        \"min\": 1.0,\n        \"max\": 6.9,\n        \"num_unique_values\": 43,\n        \"samples\": [\n          6.7,\n          3.8,\n          3.7\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"PetalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.7631607417008411,\n        \"min\": 0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n        \"samples\": [\n          0.2,\n          1.2,\n          1.3\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Species\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 3,\n        \"samples\": [\n          \"Iris-setosa\",\n          \"Iris-versicolor\",\n          \"Iris-virginica\"\n        ],\n        \"semantic_type\": \"\",
```

```
\n"description\\": \\\"\\\"\\n      }\n    }\n  ]\n  }\n  \", \"type\": \"dataframe\", \"variable_name\": \"df\"}
```

```
df.tail()
```

```
{\"summary\": \"{\\n  \\\"name\\\": \\\"df\\\",\\n  \\\"rows\\\": 5,\\n  \\\"fields\\\": [\\n  
  {\\n    \\\"column\\\": \\\"Id\\\",\\n    \\\"properties\\\": {\\n  
    \\\"dtype\\\": \\\"number\\\",\\n    \\\"std\\\": 1,\\n    \\\"min\\\": 146,\\n  
    \\\"max\\\": 150,\\n    \\\"num_unique_values\\\": 5,\\n    \\\"samples\\\":  
    [\\n      147,\\n      150,\\n      148\\n    ],\\n  
    \\\"semantic_type\\\": \\\"\\\",\\n    \\\"description\\\": \\\"\\\"\\n    }\\n  
    },\\n    {\\n      \\\"column\\\": \\\"SepalLengthCm\\\",\\n  
    \\\"properties\\\": {\\n      \\\"dtype\\\": \\\"number\\\",\\n      \\\"std\\\":  
    0.30331501776206193,\\n      \\\"min\\\": 5.9,\\n      \\\"max\\\": 6.7,\\n  
    \\\"num_unique_values\\\": 5,\\n      \\\"samples\\\": [\\n        6.3,\\n  
    5.9,\\n        6.5\\n      ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n  
    \\\"description\\\": \\\"\\\"\\n    }\\n    },\\n    {\\n      \\\"column\\\":  
    \\\"SepalWidthCm\\\",\\n      \\\"properties\\\": {\\n      \\\"dtype\\\":  
    \\\"number\\\",\\n      \\\"std\\\": 0.31937438845342625,\\n      \\\"min\\\":  
    2.5,\\n      \\\"max\\\": 3.4,\\n      \\\"num_unique_values\\\": 3,\\n  
    \\\"samples\\\": [\\n        3.0,\\n        2.5,\\n        3.4\\n  
    ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n      \\\"description\\\": \\\"\\\"\\n  
    }\\n    },\\n    {\\n      \\\"column\\\": \\\"PetalLengthCm\\\",\\n  
    \\\"properties\\\": {\\n      \\\"dtype\\\": \\\"number\\\",\\n      \\\"std\\\":  
    0.14832396974191348,\\n      \\\"min\\\": 5.0,\\n      \\\"max\\\": 5.4,\\n  
    \\\"num_unique_values\\\": 4,\\n      \\\"samples\\\": [\\n        5.0,\\n  
    5.1,\\n        5.2\\n      ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n  
    \\\"description\\\": \\\"\\\"\\n    }\\n    },\\n    {\\n      \\\"column\\\":  
    \\\"PetalWidthCm\\\",\\n      \\\"properties\\\": {\\n      \\\"dtype\\\":  
    \\\"number\\\",\\n      \\\"std\\\": 0.23021728866442667,\\n      \\\"min\\\":  
    1.8,\\n      \\\"max\\\": 2.3,\\n      \\\"num_unique_values\\\": 4,\\n  
    \\\"samples\\\": [\\n        1.9,\\n        1.8,\\n        2.3\\n  
    ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n      \\\"description\\\": \\\"\\\"\\n  
    }\\n    },\\n    {\\n      \\\"column\\\": \\\"Species\\\",\\n  
    \\\"properties\\\": {\\n      \\\"dtype\\\": \\\"category\\\",\\n  
    \\\"num_unique_values\\\": 1,\\n      \\\"samples\\\": [\\n        \\\"Iris-  
    virginica\\\"\\n      ],\\n      \\\"semantic_type\\\": \\\"\\\",\\n  
    \\\"description\\\": \\\"\\\"\\n    }\\n    }\\n  ]\\n  }\", \"type\": \"dataframe\"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 6 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   Id                     150 non-null   int64  
1   SepalLengthCm          150 non-null   float64  
2   SepalWidthCm           150 non-null   float64  
3   PetalLengthCm          150 non-null   float64
```

```
4   PetalWidthCm    150 non-null    float64
5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
df.describe()
```

```
{
  "summary": {
    "name": "df",
    "rows": 8,
    "fields": [
      {
        "column": "Id",
        "properties": {
          "dtype": "number",
          "std": 53.756293020494844,
          "min": 1.0,
          "max": 150.0,
          "num_unique_values": 6,
          "samples": [
            150.0,
            75.5,
            112.75
          ],
          "semantic_type": "",
          "description": ""
        }
      },
      {
        "column": "SepalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 51.24711349471842,
          "min": 0.828066127977863,
          "max": 150.0,
          "num_unique_values": 8,
          "samples": [
            5.8433333333333334,
            5.8,
            150.0
          ],
          "semantic_type": "",
          "description": ""
        }
      },
      {
        "column": "SepalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 52.08647211421483,
          "min": 0.4335943113621737,
          "max": 150.0,
          "num_unique_values": 8,
          "samples": [
            3.0540000000000003,
            3.0,
            150.0
          ],
          "semantic_type": "",
          "description": ""
        }
      },
      {
        "column": "PetalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 51.835227940958106,
          "min": 1.0,
          "max": 150.0,
          "num_unique_values": 8,
          "samples": [
            3.7586666666666666,
            4.35,
            150.0
          ],
          "semantic_type": "",
          "description": ""
        }
      },
      {
        "column": "PetalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 52.636634243409915,
          "min": 0.1,
          "max": 150.0,
          "num_unique_values": 8,
          "samples": [
            1.1986666666666668,
            1.3,
            150.0
          ],
          "semantic_type": "",
          "description": ""
        }
      }
    ],
    "type": "dataframe"
  }
}
```

```
df.isnull().sum()
```

```
Id                0
SepalLengthCm     0
SepalWidthCm      0
PetalLengthCm     0
PetalWidthCm      0
Species           0
dtype: int64
```

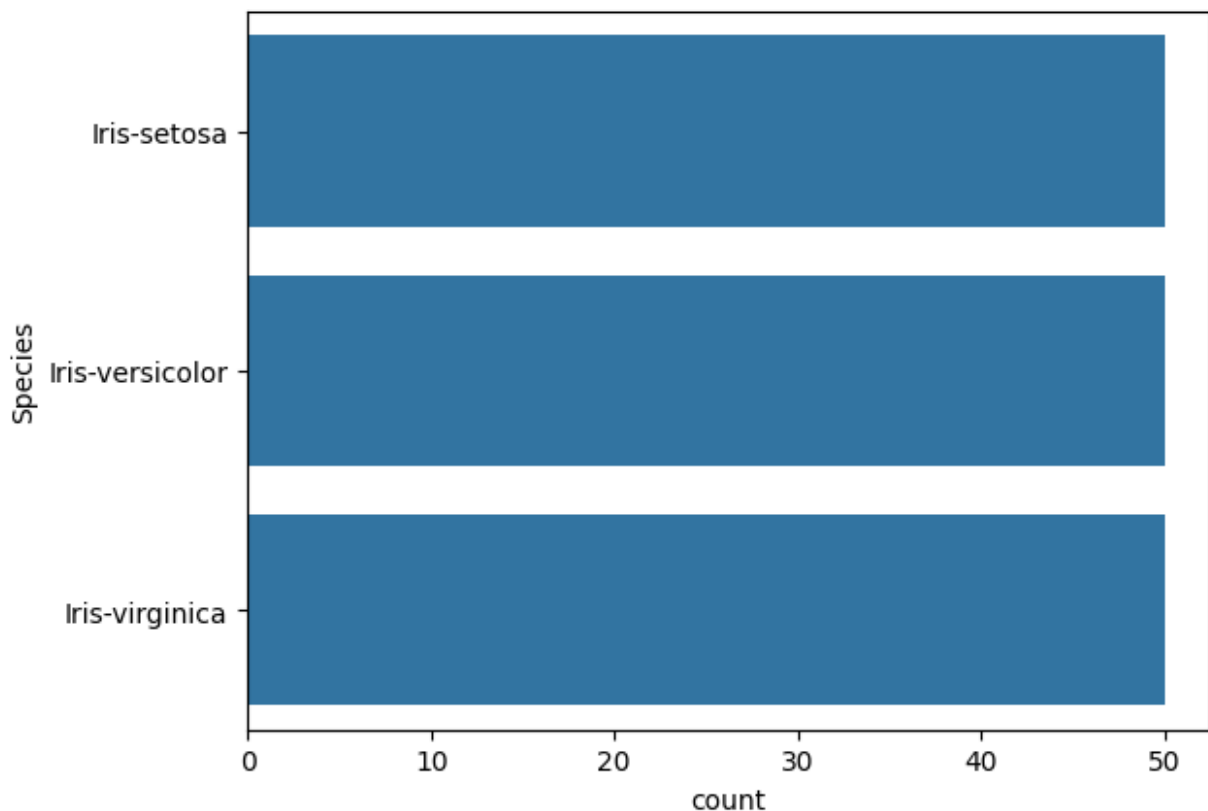
```
print(df['Species'].value_counts())
sns.countplot(df['Species'])
```

Species

Iris-setosa	50
Iris-versicolor	50
Iris-virginica	50

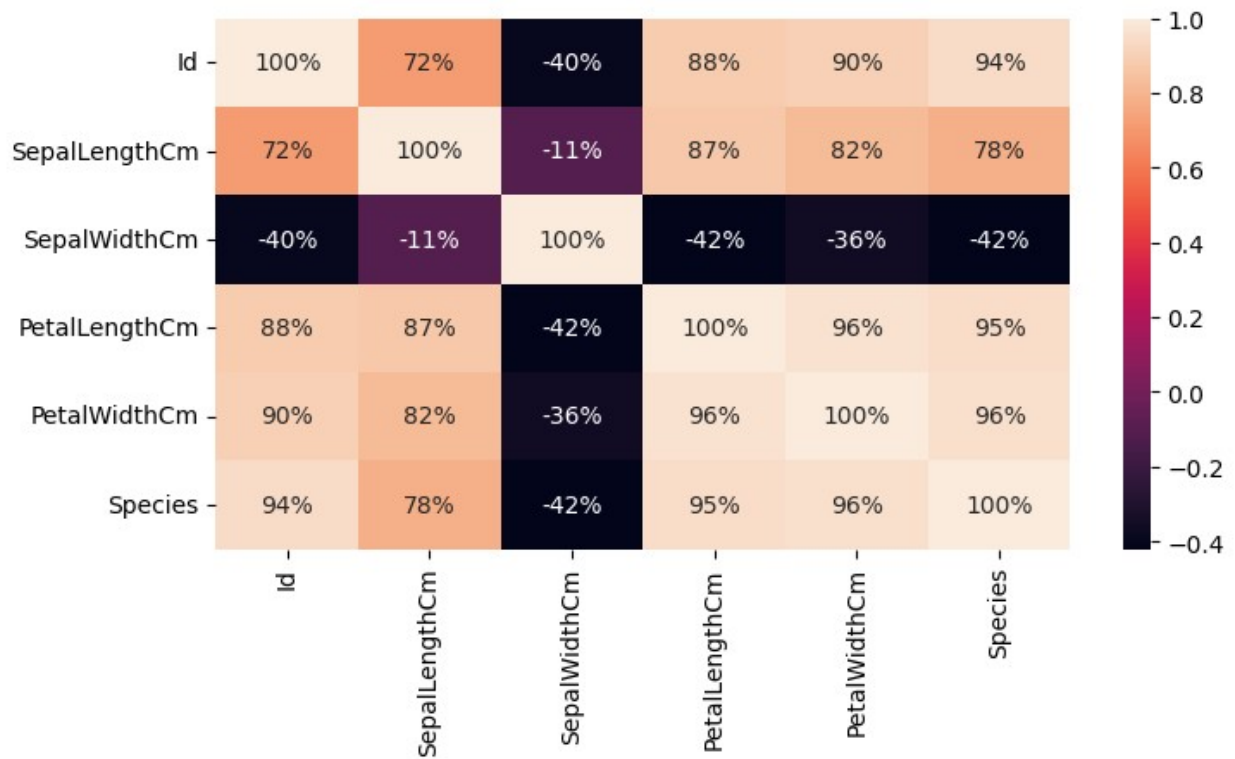
Name: count, dtype: int64

<Axes: xlabel='count', ylabel='Species'>

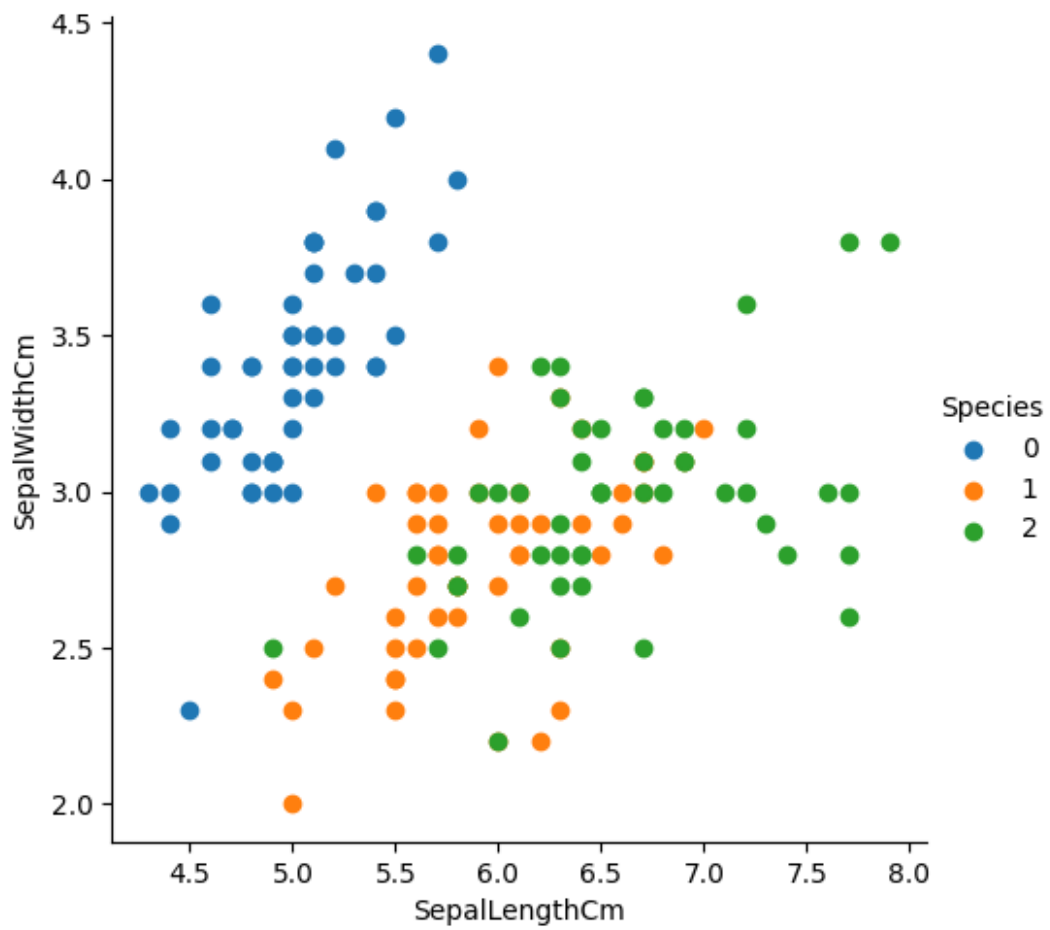


```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Species'] = le.fit_transform(df['Species'])

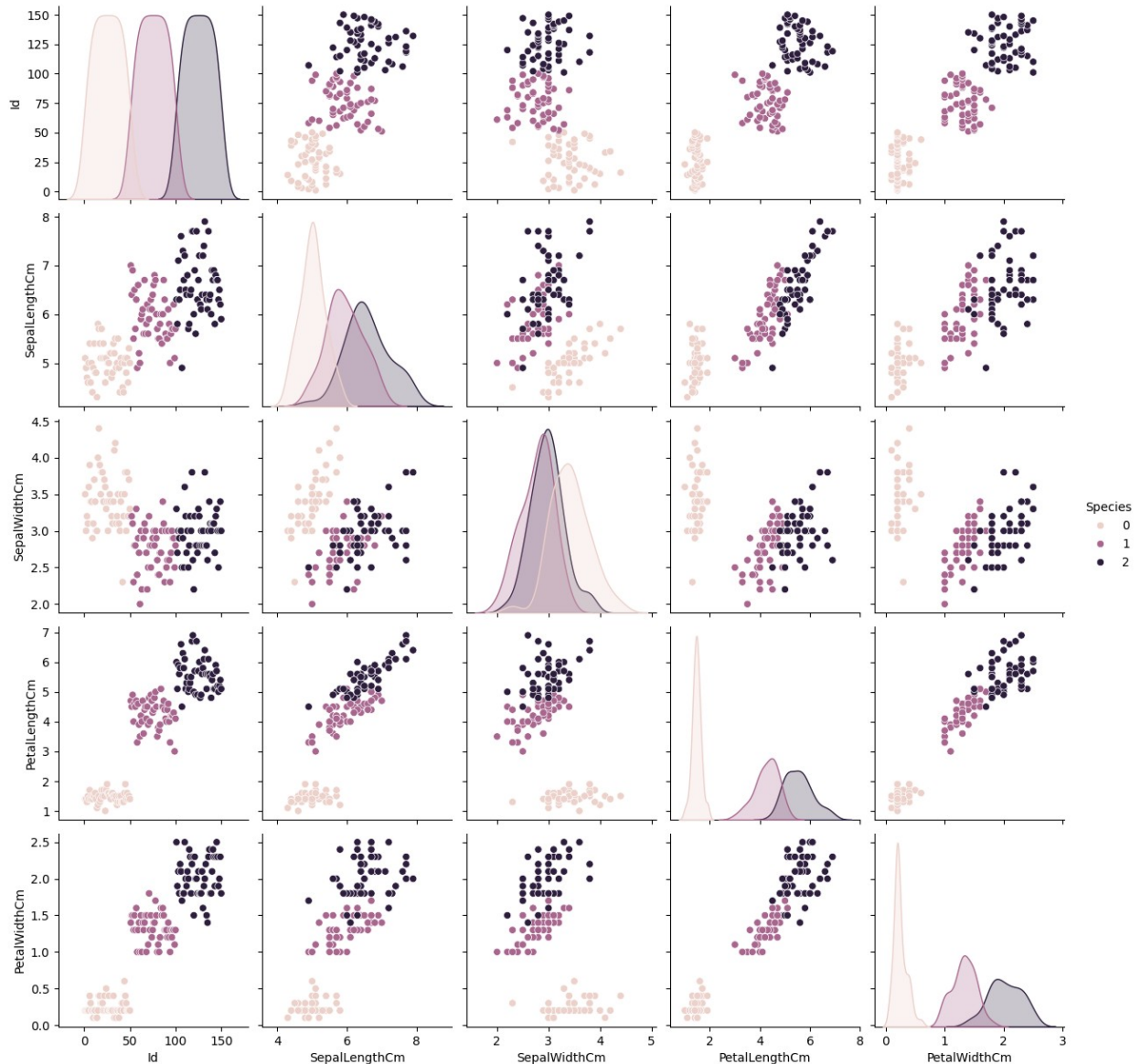
plt.figure(figsize=(8,4))
sns.heatmap(df.corr(), annot=True, fmt=".0%")
plt.show()
```



```
sns.FacetGrid(df, hue="Species",
height=5).map(plt.scatter,"SepalLengthCm",
"SepalWidthCm").add_legend()
<seaborn.axisgrid.FacetGrid at 0x7a92858ae9b0>
```



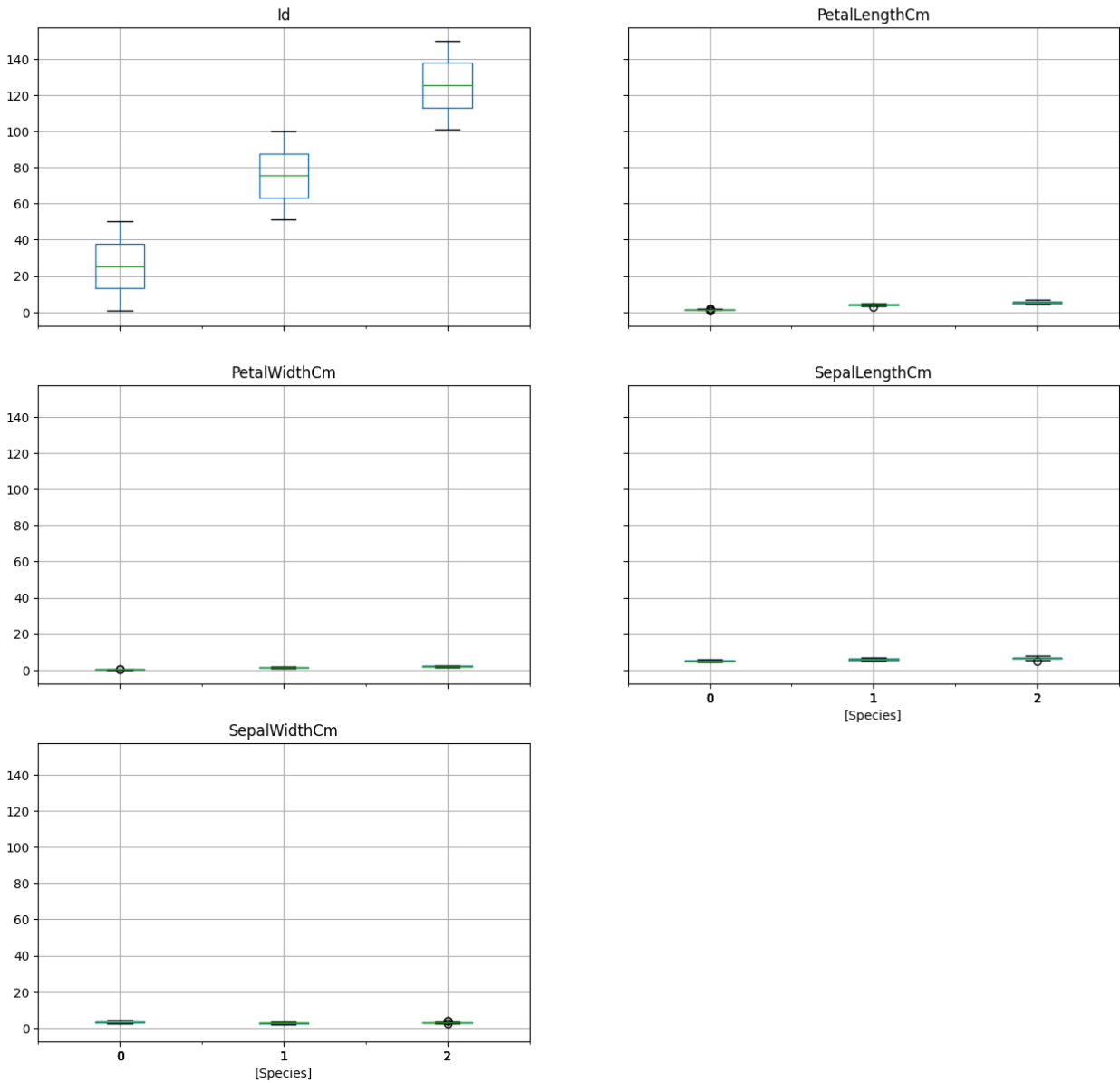
```
sns.pairplot(df.iloc[:, :], hue='Species')  
<seaborn.axisgrid.PairGrid at 0x7a92846a7eb0>
```



```
df.boxplot(by="Species", figsize=(15,15))

array([[<Axes: title={'center': 'Id'}, xlabel='[Species] '>,
        <Axes: title={'center': 'PetalLengthCm'},
        xlabel='[Species] '>],
       [<Axes: title={'center': 'PetalWidthCm'}, xlabel='[Species] '>,
        <Axes: title={'center': 'SepalLengthCm'},
        xlabel='[Species] '>],
       [<Axes: title={'center': 'SepalWidthCm'}, xlabel='[Species] '>,
        <Axes: >]], dtype=object)
```

Boxplot grouped by Species



df

```

{"summary": "{\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\n    {\n      \"column\": \"Id\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 43,\n        \"min\": 1,\n        \"max\": 150,\n        \"num_unique_values\": 150,\n        \"samples\": [\n          74,\n          19,\n          119\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"SepalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.828066127977863,\n        \"min\":

```



```

4.3,\n          \"max\": 7.9,\n          \"num_unique_values\": 35,\n\"samples\": [\n          6.2,\n          4.5,\n          5.6\n],\n          \"semantic_type\": \"\",\n          \"description\": \"\"\n}\n    },\n    {\n        \"column\": \"SepalWidthCm\",\n        \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0.4335943113621737,\n            \"min\": 2.0,\n            \"max\": 4.4,\n            \"num_unique_values\": 23,\n            \"samples\": [\n                2.3,\n                4.0,\n                3.5\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"PetalLengthCm\",\n        \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 1.7644204199522626,\n            \"min\": 1.0,\n            \"max\": 6.9,\n            \"num_unique_values\": 43,\n            \"samples\": [\n                6.7,\n                3.8,\n                3.7\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"PetalWidthCm\",\n        \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0.7631607417008411,\n            \"min\": 0.1,\n            \"max\": 2.5,\n            \"num_unique_values\": 22,\n            \"samples\": [\n                0.2,\n                1.2,\n                1.3\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"Species\",\n        \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0,\n            \"min\": 0,\n            \"max\": 2,\n            \"num_unique_values\": 3,\n            \"samples\": [\n                0,\n                1,\n                2\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n        }\n    }\n]\n}\n\"type\": \"dataframe\", \"variable_name\": \"df\"}

```

df.shape

(150, 6)

```
X = df.iloc[:,0:4]
```

```
y = df['Species']
```

X

```

{"summary": "{\n  \"name\": \"X\",\n  \"rows\": 150,\n  \"fields\": [\n    {\n      \"column\": \"Id\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 43,\n        \"min\": 1,\n        \"max\": 150,\n        \"num_unique_values\": 150,\n        \"samples\": [\n          74,\n          19,\n          119\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"SepalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.828066127977863,\n        \"min\": 4.3,\n        \"max\": 7.9,\n        \"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n          4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"SepalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\":

```

```
0.4335943113621737,\n          \"min\": 2.0,\n          \"max\": 4.4,\n          \"num_unique_values\": 23,\n          \"samples\": [\n            2.3,\n            4.0,\n            3.5\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\"\n        },\n        {\n          \"column\":\n            \"PetalLengthCm\",\n          \"properties\": {\n            \"dtype\":\n              \"number\",\n            \"std\": 1.7644204199522626,\n            \"min\":\n              1.0,\n            \"max\": 6.9,\n            \"num_unique_values\": 43,\n            \"samples\": [\n              6.7,\n              3.8,\n              3.7\n            ],\n            \"semantic_type\": \"\",\n            \"description\": \"\"\n          }\n        }\n      ],\n      \"type\": \"dataframe\", \"variable_name\": \"X\"}
```

y

```
0    0
1    0
2    0
3    0
4    0
```

```
..
145   2
146   2
147   2
148   2
149   2
```

Name: Species, Length: 150, dtype: int64

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state=42)
```

X_train.shape

```
(120, 4)
```

X_test.shape

```
(30, 4)
```

y_train.shape

```
(120,)
```

y_test.shape

```
(30,)
```

```
log = LogisticRegression()
log.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to converge
(status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(  
LogisticRegression()  
prediction = log.predict(X_test)  
metrics.accuracy_score(prediction,y_test)  
1.0
```