AUTHOR :- KANSARA RUSHABH BHAVESH TECHNOLOGY/DOMAIN :- MACHINE LEARNING

INTRODUCTION

→ This project focuses on the Exploratory Data Analysis (EDA) of the famous Iris flower dataset, which is a foundational benchmark in the field of data science and machine learning. The primary objective is to investigate the physical properties of Iris flowers to understand how different measurements correlate with their species classification.

→ The challenge is to differentiate between three distinct species of Iris flowers (Iris setosa, Iris versicolor and Iris virginica) based solely on four physical attributes :- Sepal Length, Sepal Width, Petal Length and Petal Width.

GOAL

→ The ultimate goal of this analysis is to visually confirm which features (such as petal size) are most effective at separating the three species, setting the stage for building an accurate machine learning classification model.

DEFINING THE CORE IDEA

→ Data-Driven Classification :- Instead of using complex biology, the project tests if simple numbers (Sepal Length, Sepal Width, Petal Length, Petal Width) are enough to identify a species.

→ Feature Importance :- It identifies which parts of the flower matter most. Your analysis (specifically the pairplot) reveals that Petal dimensions are far better at separating the species than Sepal dimensions.

→ Visual Confirmation of Separability :- The project visually proves that the data falls into distinct "clusters." Since the dots for "Setosa" (target 0) are far apart from the others in your graphs, it confirms that a machine learning model will be able to classify them with high accuracy.

LIBRARIES USED IN THIS PROJECT

1. PANDAS
   - →Role :- Used to convert the raw numbers into a Data Frame (A table with rows and columns).
   - →Clears the readability , allows to give the names to the columns.
   - →Analysis :- we use “df.describe()” method to instantly calculate the mean, min, max and standard deviation of the data.

2. NUMPY
   - →Stands for “Numerical python”.
   - →Role :- Act as a “Calculator”. Provides “ndArray” to perform complex mathematical problems.
   - →Pandas is built on top of NumPy, so when you manipulate the data, NumPy handles the heavy mathematical lifting in the background to keep it fast.

3. MATPLOTLIB
   - →Role :- Used for the visualization of the dataset.
   - →Helps to learn the data with the help of the diagram.
   - →Base plotting :- It provides the underlying framework for all graphs.
   - →Customization :- Using “Seaborn” for the fancy plots.

4. SEABORN
   - →Role :- Using pair plot (sns.pairplot()) creates the grid of scatter plots that compares every feature against every other feature.
   - →Differentiation **:-** It automatically handles the color-coding (hue) to distinguish between the three flower species, which would be very difficult to do with just Matplotlib.

5. SCIKIT-LEARN
   - →It is the machine learning library.
   - →**Data Source:** You used datasets.load_iris() to fetch the data. This saves you from having to download a CSV file manually, as the library has the data built-in for practice.
   - →To train data for prediction.

6. OS
   - →It provides functions for interacting with the operating system
   - →Utility :- In this specific notebook, it is likely a standard "boilerplate" import (code you include in every project out of habit). Unless you are saving files to specific folders or reading external paths, it is probably not being heavily used here.

SOFTWARE REQUIREMENT SPECIFICATION

LIST	DESCRIPTION
Python	Version 3.7 or higher The syntax and libraries used (especially the f-strings or newer Pandas features if you use them later) require a modern Python version.
Development IDE	Jupyter Notebook

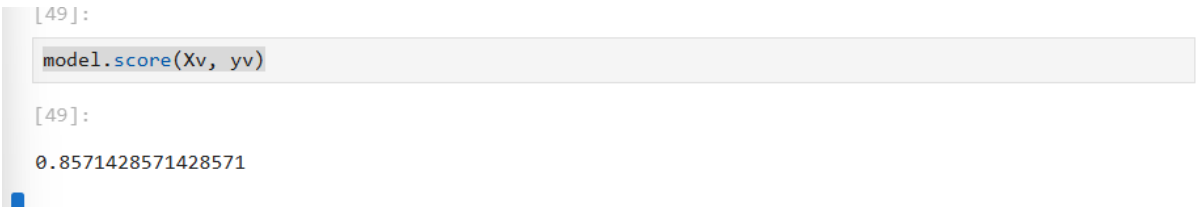
MODEL USED WITH CODE AND SCREENSHOT..

1. Logistic Regression
   - → This used to define “Category”.
   - → Since project is about classification (deciding which of the 3 species a flower belongs to), Logistic Regression would be the specific mathematical engine that draws the "boundary lines" between the groups you saw in your graphs.
   - → In this project, Logistic Regression acts as the Multi-Class Classifier. It analyses the input measurements (sepal/petal dimensions) to calculate the probability of a flower belonging to one of the three species, effectively drawing decision boundaries to separate the distinct clusters observed in the Exploratory Data Analysis.

CODE :-

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(max_iter=200)
# Xt stands for X_train and Xv stands for X_validation
Xt, Xv, yt, yv = train_test_split(x_train, y_train, test_size=0.25)
model.fit(Xt, yt)
y_pred = model.predict(Xv)
np.mean(y_pred == yv)
model.score(Xv, yv)
```

OUTPUT :-



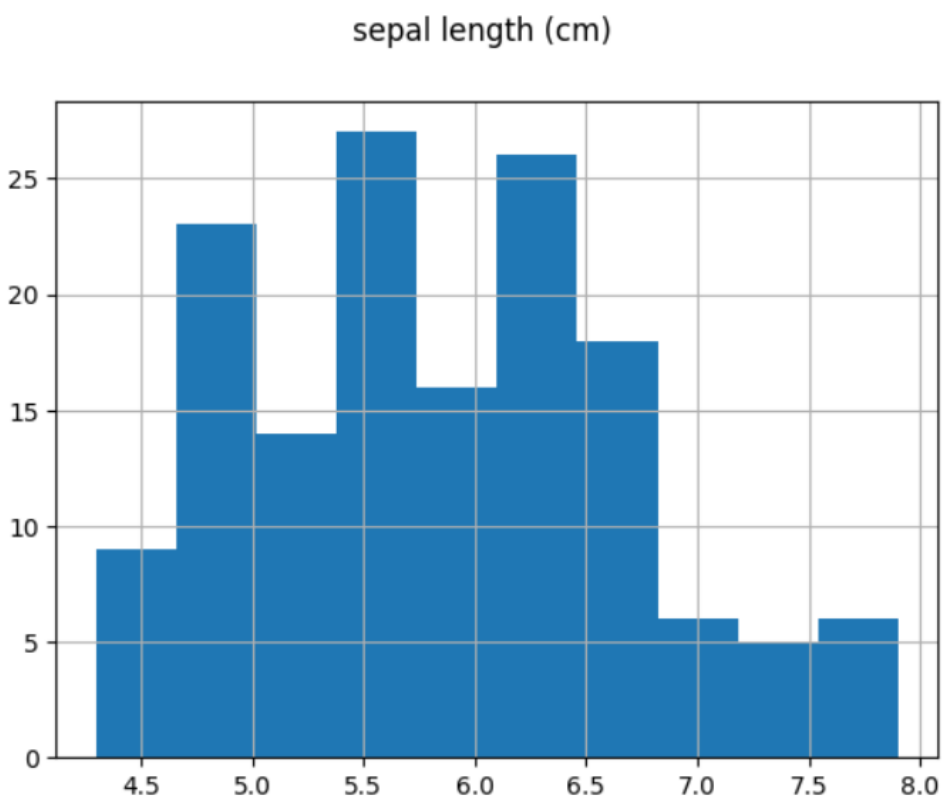
LIST AND USE OF THE DIAGRAMS

1. Histograms

→ For Univariate Analysis.
→ Helps you to see the shape of the data.
→ Outlier detection :-
→ If you saw a bar far away from the others (e.g., a 15cm petal), you would know there is an error in the data.

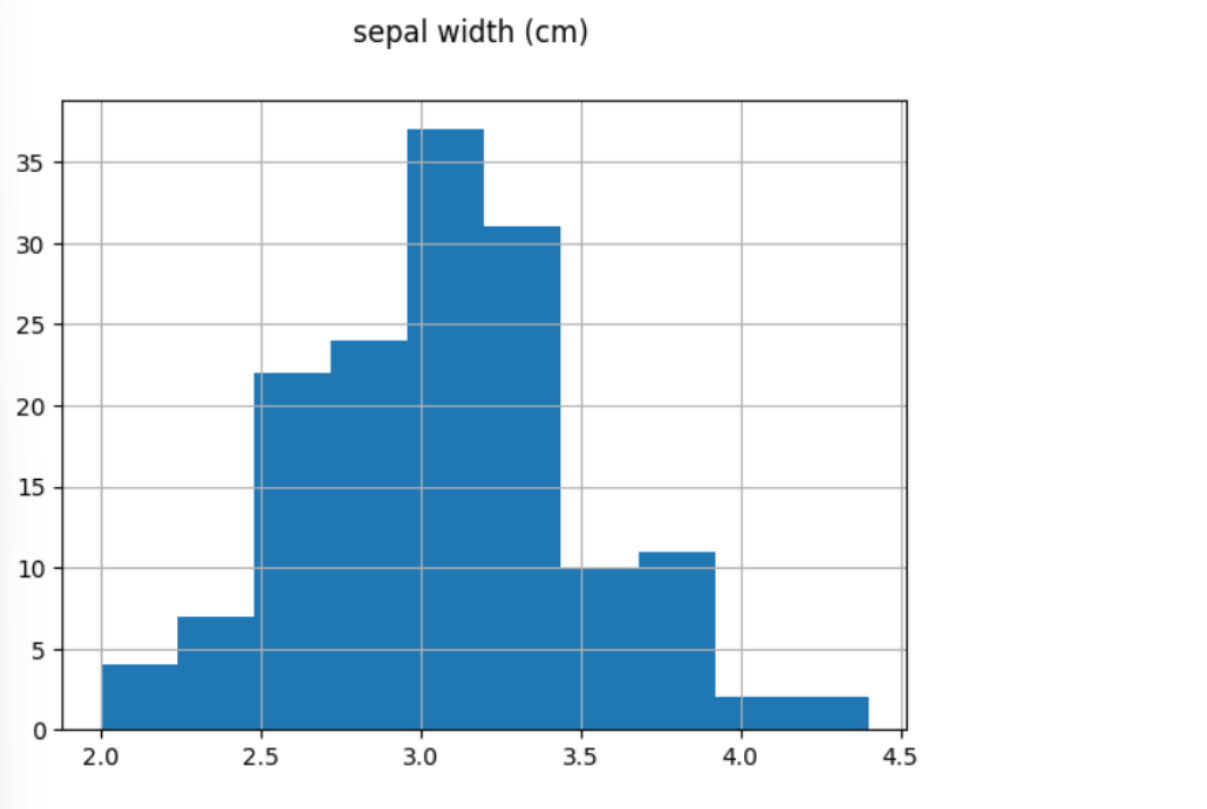
[14]:

```
col = "sepal length (cm)"
df[col].hist()
plt.suptitle(col)
plt.show()
```

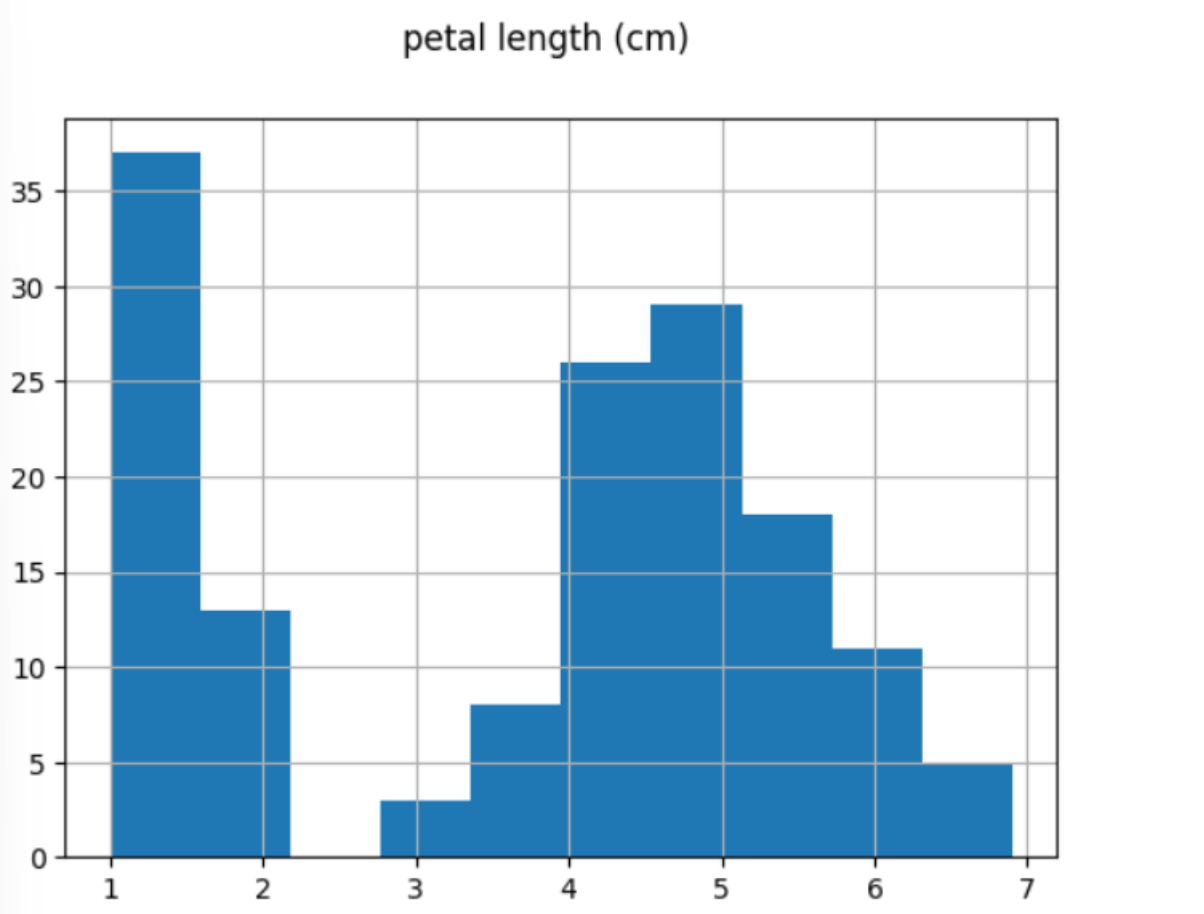


[15]:

```
col = "sepal width (cm)"
df[col].hist()
plt.suptitle(col)
plt.show()
```

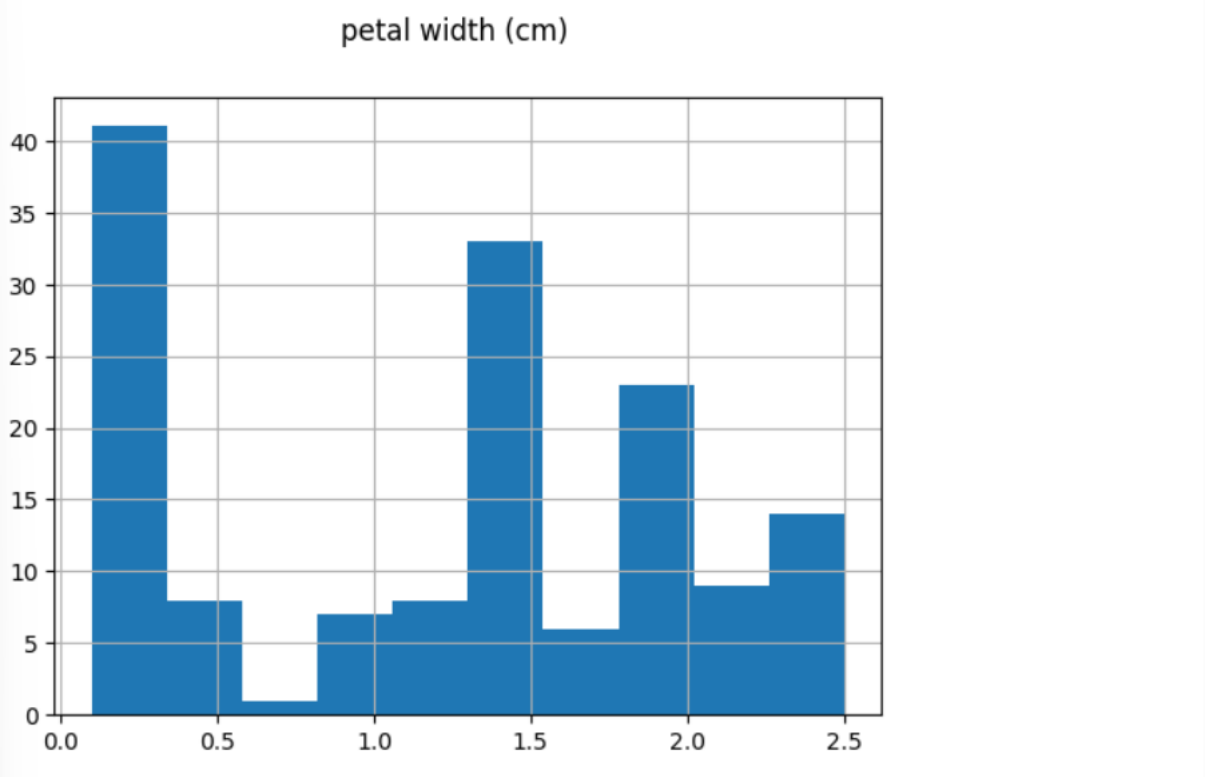


```
col = "petal length (cm)"
df[col].hist()
plt.suptitle(col)
plt.show()
```



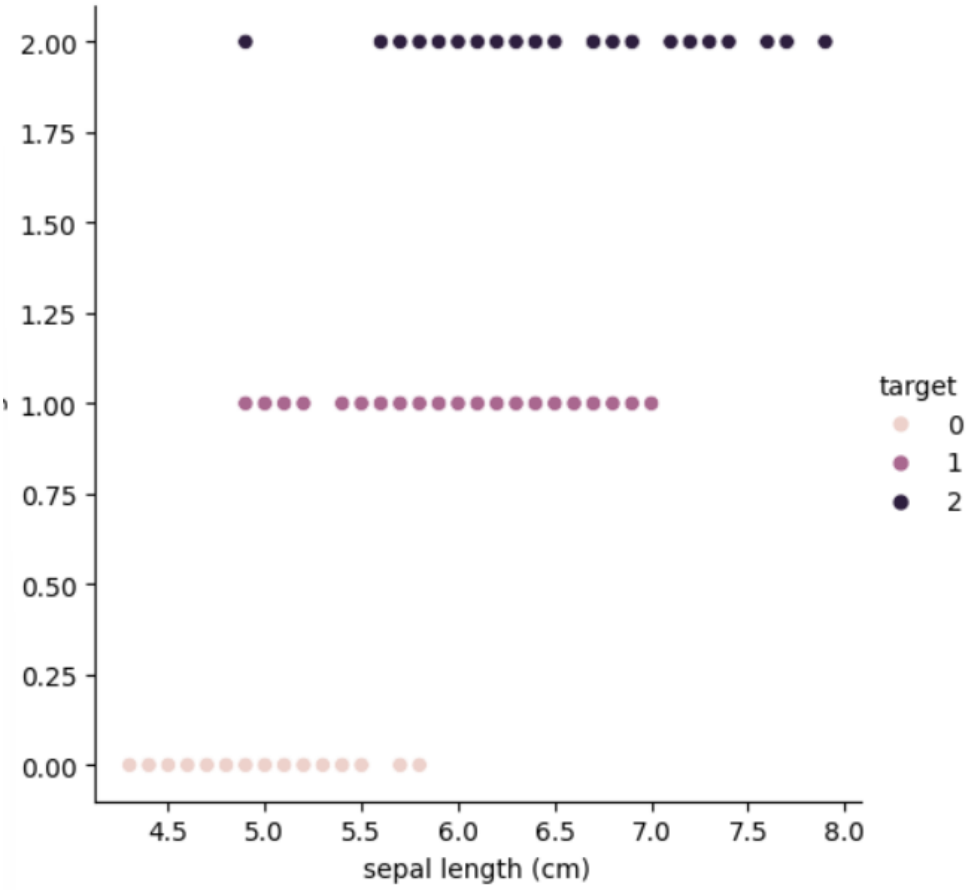
[17]:

```
col = "petal width (cm)"
df[col].hist()
plt.suptitle(col)
plt.show()
```



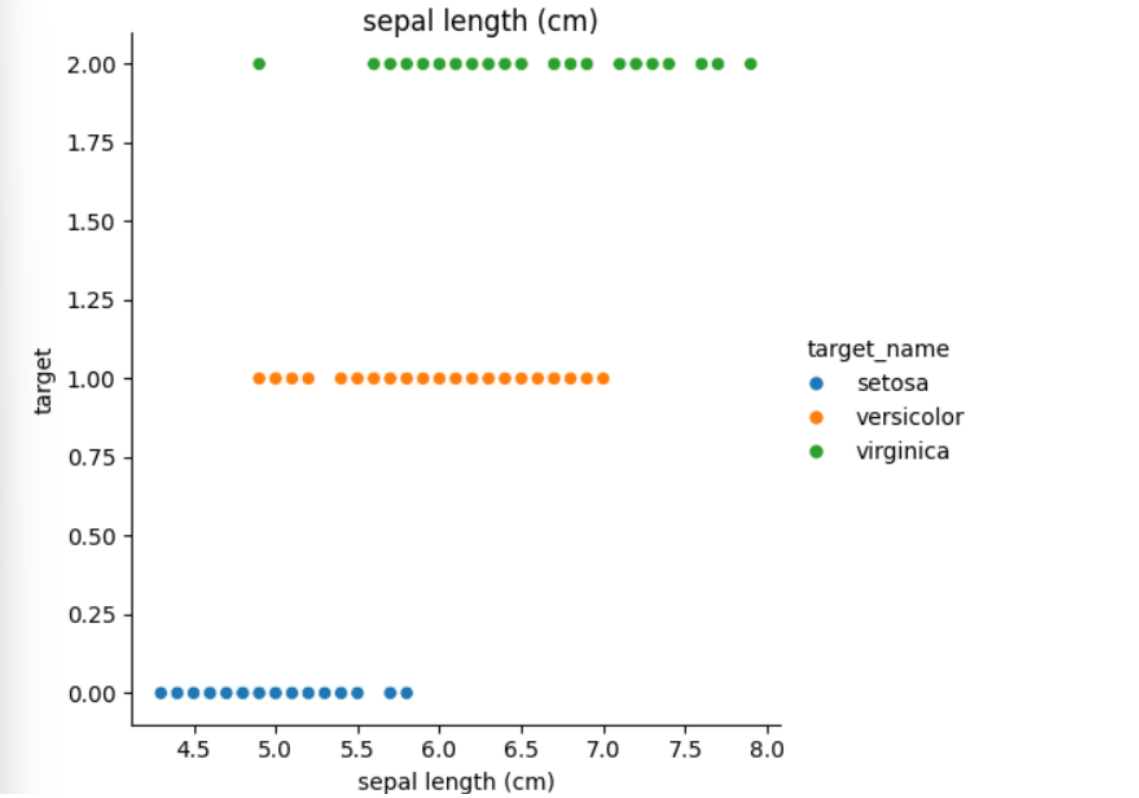
2. Seaborn Pairplot
   - → Multivariate Analysis.
   - → For Cluster classification.
   - → For feature selection.

```
eaborn.axisgrid.FacetGrid at 0x149726bd880>
```

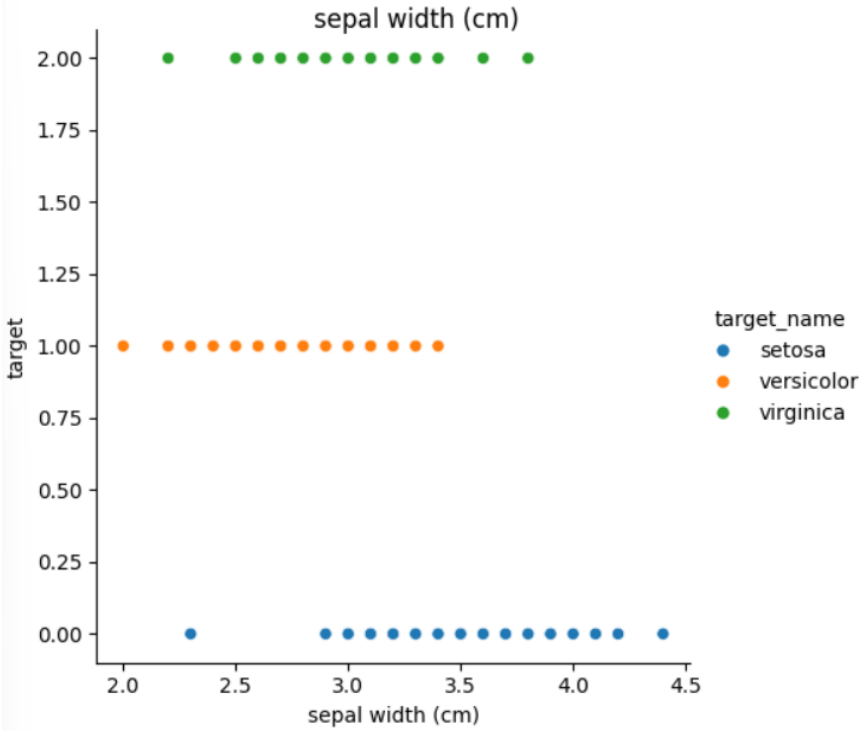


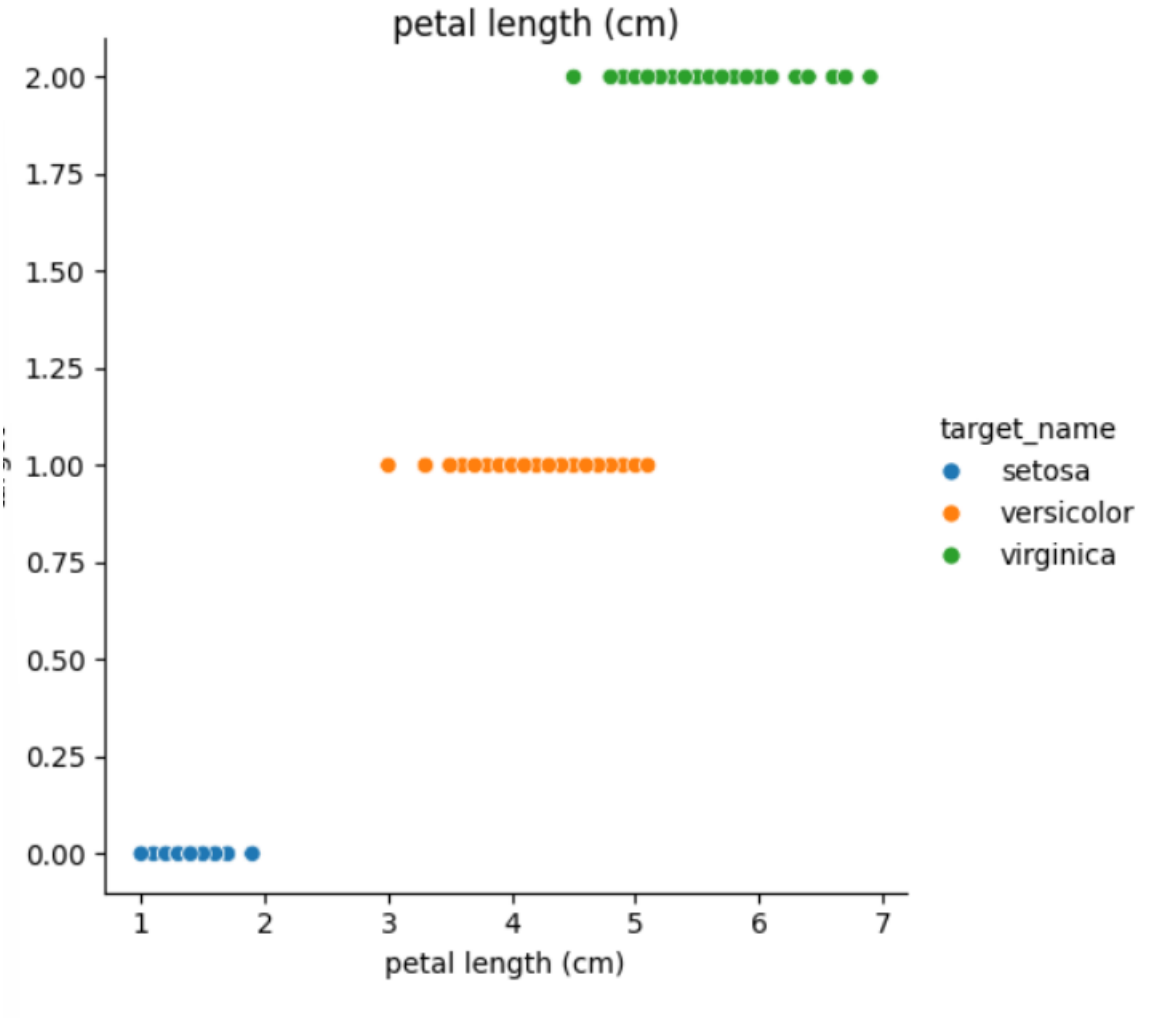
[23]:

```
col = "sepal length (cm)"
sns.relplot(x=col, y="target", hue="target_name", data=df)
plt.suptitle(col, y=1)
plt.show()
```

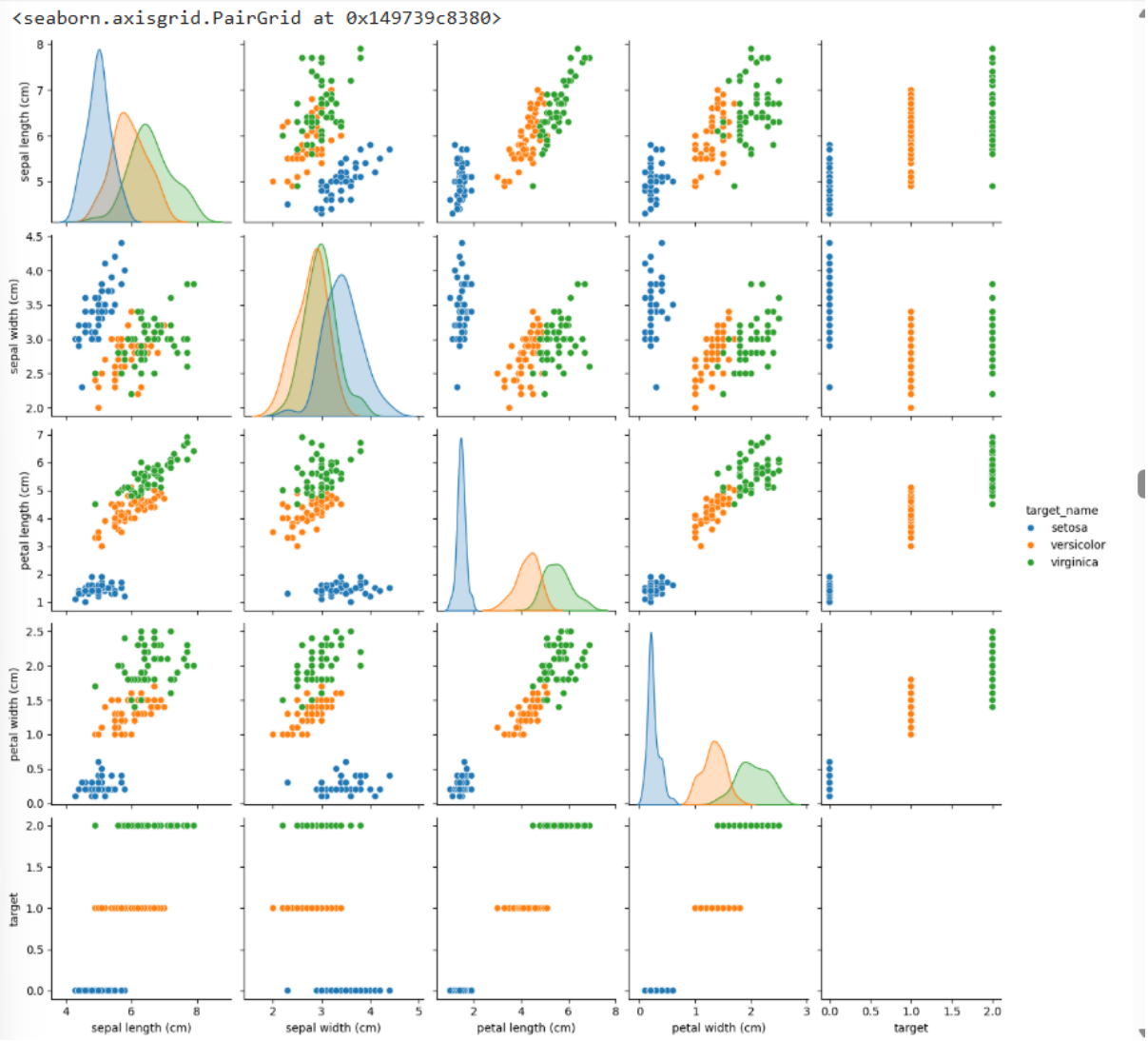


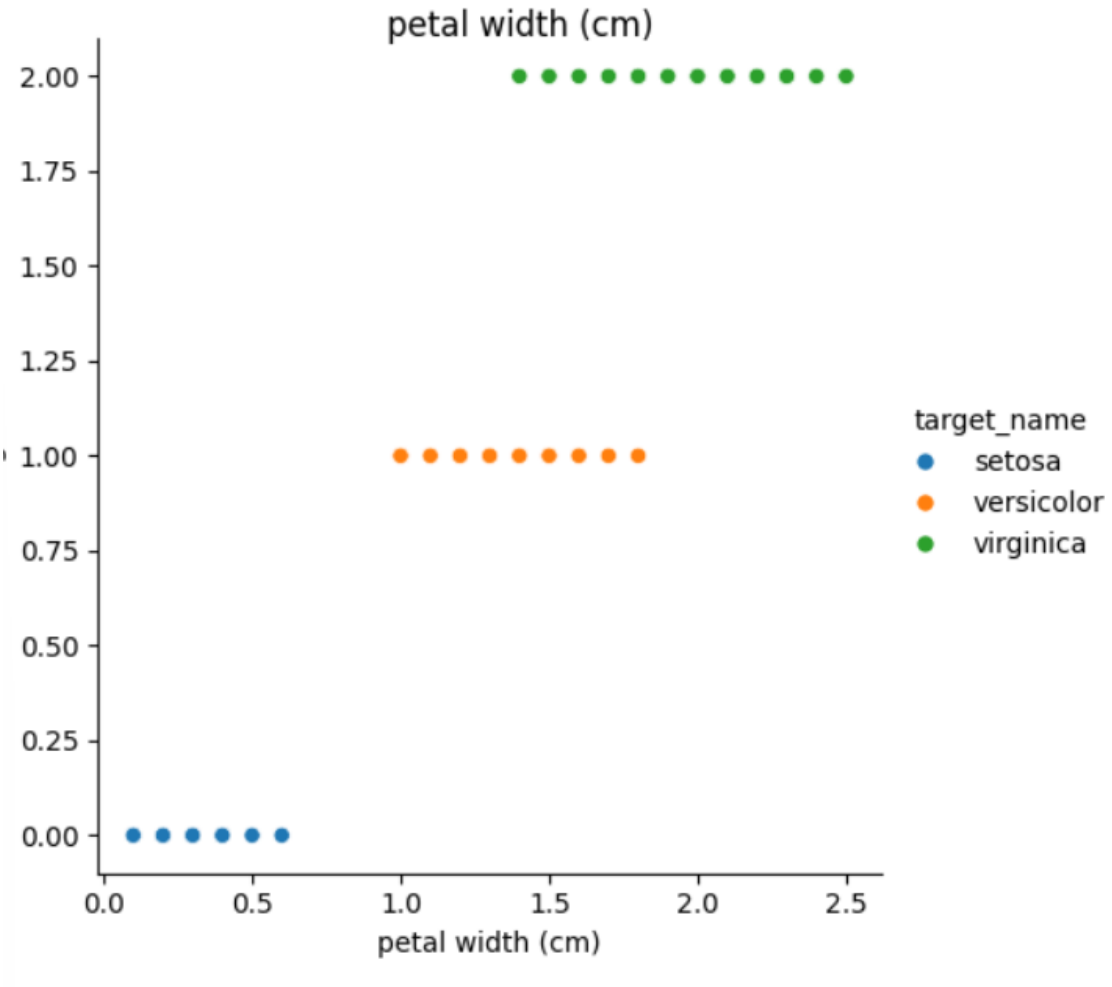
```
col = "sepal width (cm)"
sns.relplot(x=col, y="target", hue="target_name", data=df)
plt.suptitle(col, y=1)
plt.show()
```




petal length (cm)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
petal length (cm)
target_name
setosa
versicolor
virginica

AUTHOR :- KANSARA RUSHABH BHAVESH TECHNOLOGY/DOMAIN :- MACHINE LEARNING




petal width (cm)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
1.5
2.0
2.5
petal width (cm)
target_name
setosa
versicolor
virginica

CONCLUSION

This project successfully executed an Exploratory Data Analysis (EDA) on the Iris dataset, providing critical insights into the physical characteristics of Iris setosa, Iris versicolor, and Iris virginica.

FUTUTRE ENHANCEMENT

1. Model Implementation & Comparison
   - → Train multiple classifiers (Logistic Regression, Decision Trees, SVM, and Random Forest) to see which one yields the highest accuracy.
   - → Generate a performance comparison table to select the best model.

2. Hyperparameter Tuning
   - → Use techniques like GridSearchCV to find the optimal settings for the models (e.g., finding the perfect "K" in K-Nearest Neighbors or the ideal depth for a Decision Tree) to improve performance further.

3. Advanced Visualization
   - → Implement 3D Scatter Plots to visualize the clusters in a three-dimensional space (e.g., Sepal Length vs. Petal Length vs. Petal Width).
   - → Use interactive libraries like Plotly so users can zoom and rotate the graphs.

4. Deployment
   - → Build a simple user interface using Streamlit or Flask.
   - → This would allow a user to mentally enter the 4 flower measurements and get an instant prediction of the species on their screen.

5. Unsupervised Learning
   - → Apply clustering algorithms like K-Means Clustering to see if the model can identify the three species groups without being told the labels (Target column) beforehand.