

Task 3 Report

Visualizing High-Dimensional Data

1. Task Description

The task involves visualizing high-dimensional data using a **parallel coordinates plot**, a tool that helps in comparing data points across multiple dimensions. In this case, we are using the **Iris dataset**, which contains several features, including sepal length, sepal width, petal length, and petal width. The goal is to generate a parallel coordinates plot that allows us to visually analyze the relationships between different features and observe any patterns or trends in the data, particularly in relation to the target variable species.

Dataset Information

The Iris dataset is a famous dataset used in machine learning, containing 150 data points with 4 features and 1 target variable:

❖ Features:

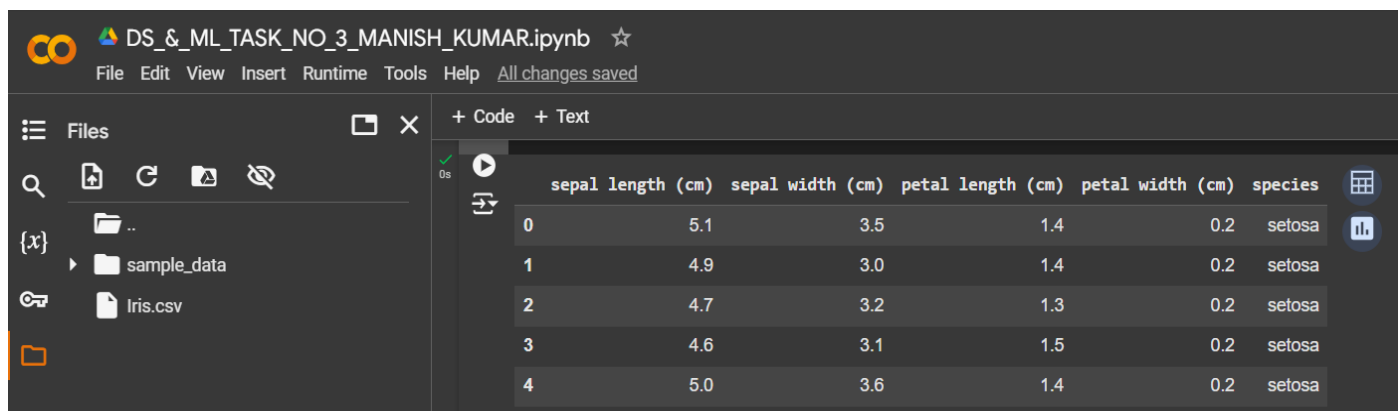
- ✚ Sepal length
- ✚ Sepal width
- ✚ Petal length
- ✚ Petal width

❖ Target variable:

- ✚ Species (Setosa, Versicolor, Virginica)

2. Attach Screenshot of Output

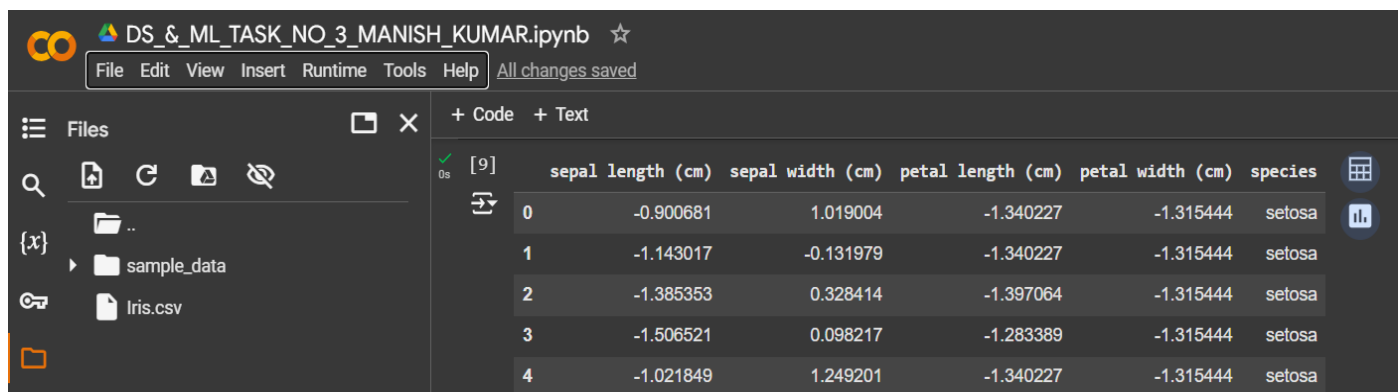
❖ Dataset Preview:



The screenshot shows a Jupyter Notebook interface with the file 'DS_& ML_TASK_NO_3_MANISH_KUMAR.ipynb'. The left sidebar shows a file explorer with 'sample_data' and 'Iris.csv'. The main area displays a table with 6 columns: 'sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)', and 'species'. The table contains 5 rows of data, all of which are 'setosa' species.

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

❖ Scaled data:



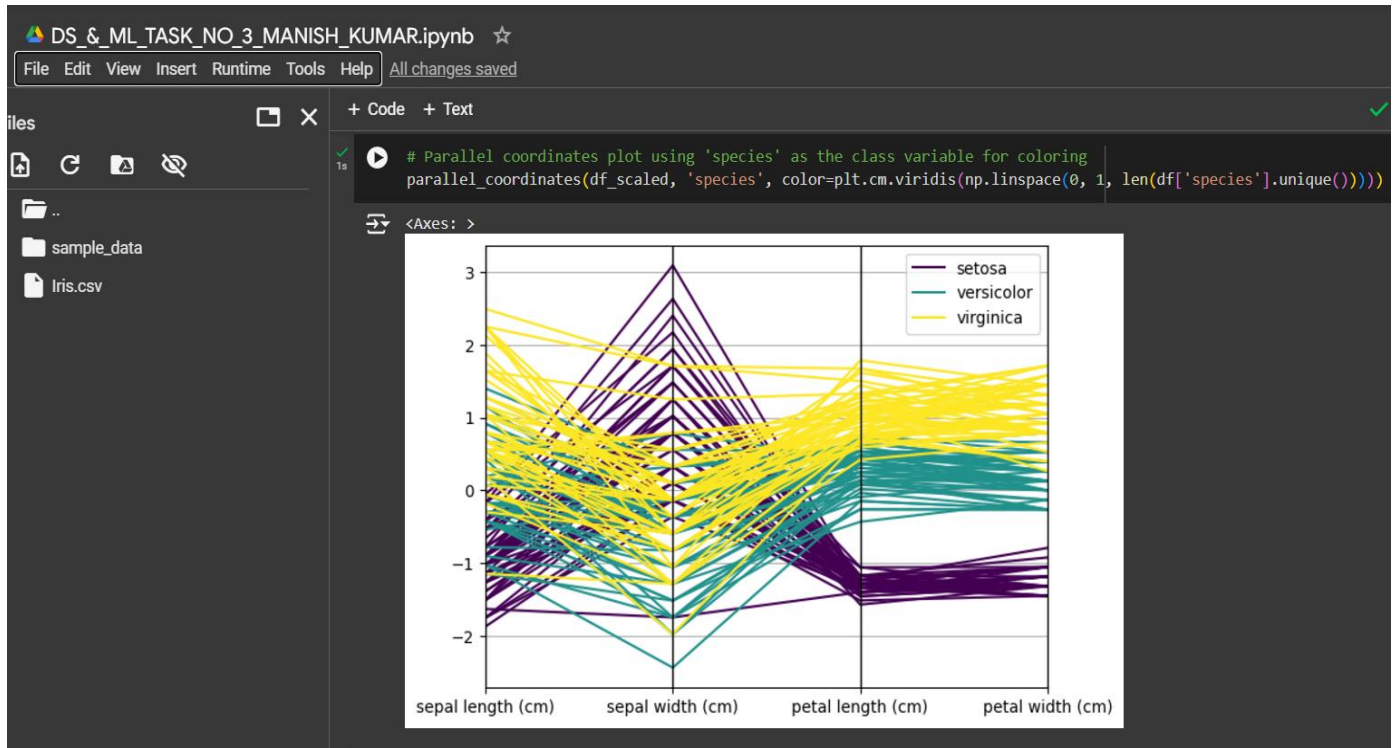
The screenshot shows the same Jupyter Notebook interface, but the data is now scaled. The table has 6 columns: 'sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)', and 'species'. The table contains 5 rows of data, all of which are 'setosa' species. The numerical values for the features are now centered around zero.

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	-0.900681	1.019004	-1.340227	-1.315444	setosa
1	-1.143017	-0.131979	-1.340227	-1.315444	setosa
2	-1.385353	0.328414	-1.397064	-1.315444	setosa
3	-1.506521	0.098217	-1.283389	-1.315444	setosa
4	-1.021849	1.249201	-1.340227	-1.315444	setosa

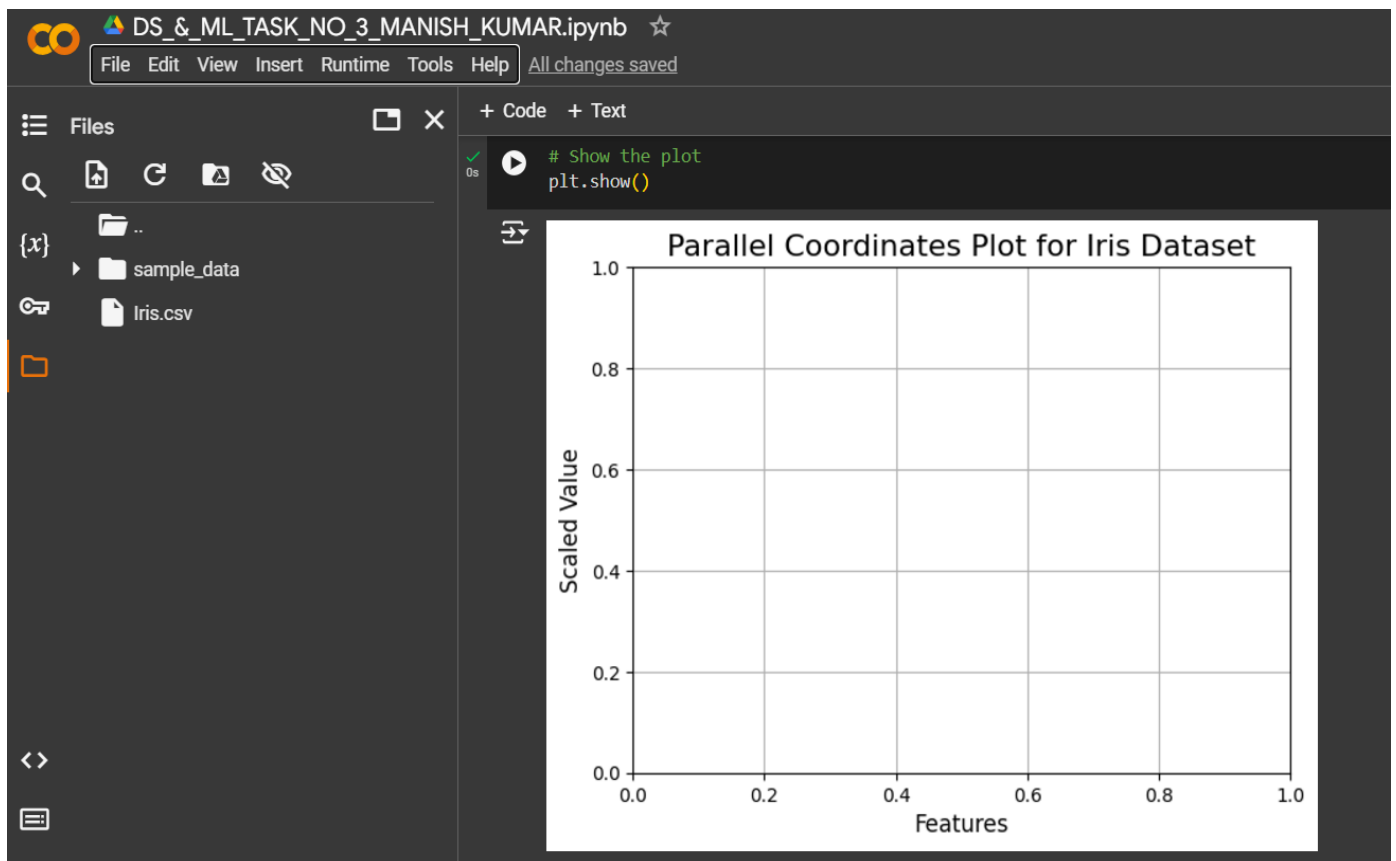
Task 3 Report

Visualizing High-Dimensional Data

❖ Parallel coordinates plot using 'species'.



❖ Parallel Coordinates Plot for Iris Dataset:



Task 3 Report

Visualizing High-Dimensional Data

3. Describe Widget/Algorithm Used in Task

❖ Parallel Coordinates Plot

- ✚ **Widget/Algorithm Used:** The **Parallel Coordinates Plot** is the main visualization technique used in this task.
- ✚ **Purpose:** It is used to visualize data with many dimensions (features) in a 2D space. This plot allows us to represent multiple data features along vertical axes and connect individual data points across those axes with lines, making it easier to analyze patterns and relationships.

How It Works:

- ✚ Each vertical axis represents a feature in the dataset.
- ✚ The data points are plotted as lines connecting their values across each of the vertical axes.
- ✚ By coloring the lines based on a class label (in this case, the species), we can visually separate and compare the different categories or classes in the dataset.

❖ StandardScaler (Data Preprocessing)

- ✚ **Widget/Algorithm Used:** The **StandardScaler** from scikit-learn is used for feature scaling in this task.
- ✚ **Purpose:** It standardizes the feature values by removing the mean and scaling to unit variance. This ensures that all features are on the same scale and prevents one feature with a larger range from dominating the plot.

How It Works:

- ✚ The **StandardScaler** subtracts the mean of each feature and divides by its standard deviation. This results in a dataset where each feature has a mean of 0 and a standard deviation of 1.

❖ Pandas and Matplotlib

- ✚ **Widget/Algorithm Used:** **Pandas** and **Matplotlib** are used for data manipulation and visualization, respectively.
- ✚ **Purpose:** Pandas is used for handling and manipulating the dataset, while Matplotlib is used to generate the plot and display it.

How It Works:

- ✚ **Pandas** provides data structures like DataFrames that make it easy to clean, preprocess, and handle data, as well as create and manipulate columns like the species column.
- ✚ **Matplotlib** is used for visualization, and specifically, the `plt.figure()` and `plt.show()` functions are used to set up and display the plot.

❖ Parallel Coordinates from Pandas Plotting

- ✚ **Widget/Algorithm Used:** The `parallel_coordinates` function from **pandas.plotting** is used to create the parallel coordinates plot.
- ✚ **Purpose:** This function generates a plot that can be used to visualize multi-dimensional data by plotting data points as lines across multiple parallel axes.

Task 3 Report

Visualizing High-Dimensional Data

How It Works:

- ✚ The `parallel_coordinates()` function takes in a DataFrame, the name of the class column (in this case, species), and a color palette to color the data points according to their class.
- ✚ The function then plots a line for each data point, where the x-axis represents different features (after scaling) and the y-axis represents the feature values.

Each of these components plays an essential role in successfully visualizing the Iris dataset and generating valuable insights into the relationships between the various features in the dataset.

Libraries/Tools Used

The following libraries were used to implement the parallel coordinates plot:

- ❖ **Pandas:** For data manipulation, cleaning, and preprocessing.
- ❖ **Matplotlib:** For generating visualizations, including the parallel coordinates plot.
- ❖ **Scikit-learn:** To load the Iris dataset and perform standardization on the features.
- ❖ **NumPy:** For numerical operations and creating color maps.
- ❖ **Pandas Plotting:** For creating the parallel coordinates plot.

*** The End ***