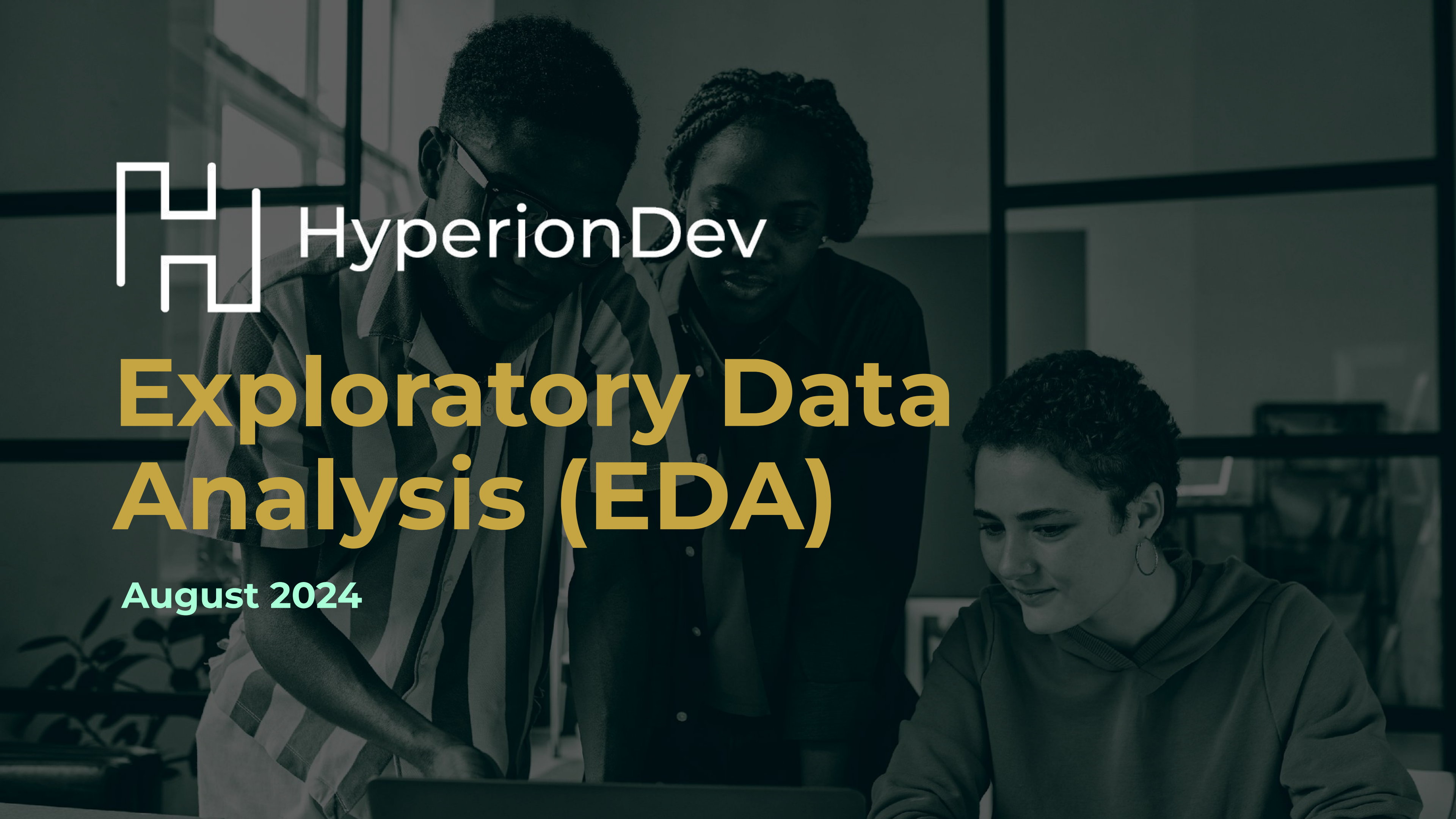


HyperionDev

Exploratory Data Analysis (EDA)

August 2024

Data Science Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Data Science Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**

- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**

- We would love your **feedback** on lectures: **Feedback on Lectures**

Problem Statements

After cleaning and preprocessing a dataset, the dataset probably contains multiple features. Before diving into modelling, it’s crucial to explore this dataset to identify key patterns, understand feature distributions, and detect any correlations between variables. By performing EDA, you can determine which features are most relevant, understand the relationships between different variables, and prepare the data for more advanced analyses.

Learning Outcomes

Understand and apply Exploratory Data Analysis (EDA) techniques to effectively analyse datasets.

- ❖ Understand the importance of EDA in DS projects
- ❖ Apply EDA techniques to clean, preprocess, & explore data
- ❖ Use Python libraries for EDA tasks and data visualisation

CoGrammar

Introduction to EDA

- **Definition:** Exploratory Data Analysis (EDA) is the process of investigating and understanding a dataset through visual and statistical techniques.
- **Importance:** EDA is a crucial first step in any data science project as it helps uncover patterns, anomalies, and relationships in the data, guiding further analysis and decision-making.

CoGrammar

Introduction to EDA

❖ **Role in the data science workflow:** EDA is performed after data collection and before model building and evaluation. It helps in understanding the data, identifying data quality issues, and selecting relevant features for modeling.

CoGrammar

Simple EDA Framework

➢ **Explore Relationships:**

- Analyse relationships between features and the target variable
- Use visualisations like scatter plots, pair plots, and correlation matrices
- Identify patterns, trends, and clusters in the data

CoGrammar

Simple EDA Framework

- **Assess Feature Importance:**
  - Determine the significance of features using statistical tests
  - Use techniques like Decision Trees or Random Forests to evaluate feature importance
  - Select relevant features based on their importance and domain knowledge

CoGrammar

Simple EDA Framework

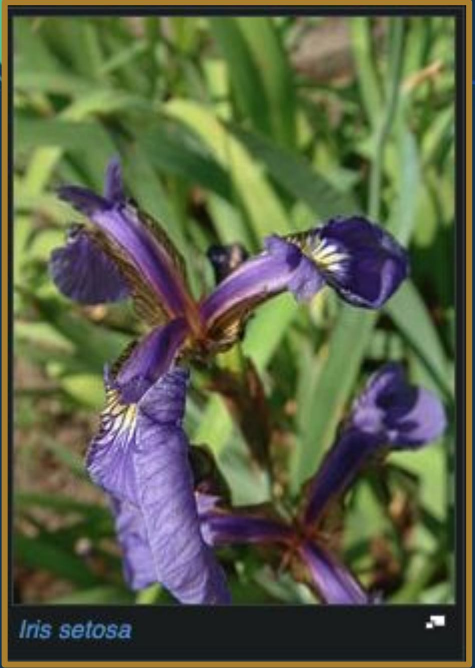
- **Iterate and Refine:**
  - Iterate on the analysis based on the insights gained
  - Refine the data cleaning and preprocessing steps if necessary
  - Consider additional visualisations or techniques to deepen the understanding of the data

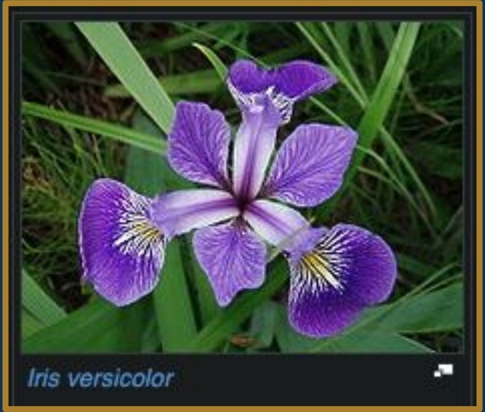
CoGrammar

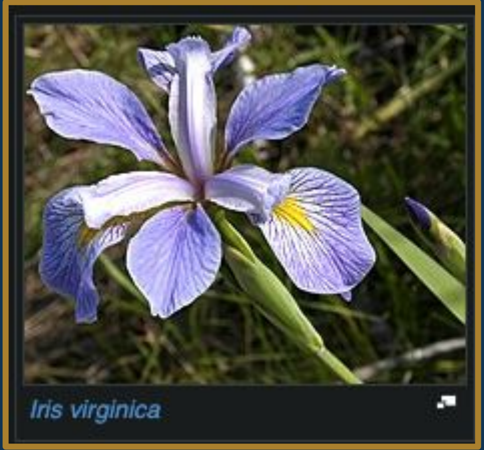
Loading and Exploring the Dataset

- To demonstrate EDA techniques, we'll use the Iris dataset from scikit-learn.
- The Iris dataset consists of measurements of sepal length, sepal width, petal length, and petal width for three species of Iris flowers.
- We'll load the dataset using scikit-learn and create a pandas DataFrame to work with.

CoGrammar

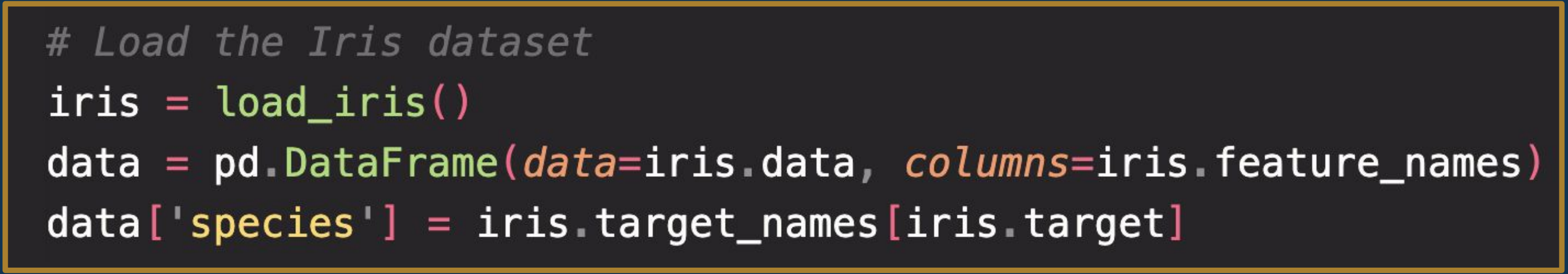

Iris setosa


Iris versicolor


Iris virginica

Source: [Wikipedia]()

CoGrammar



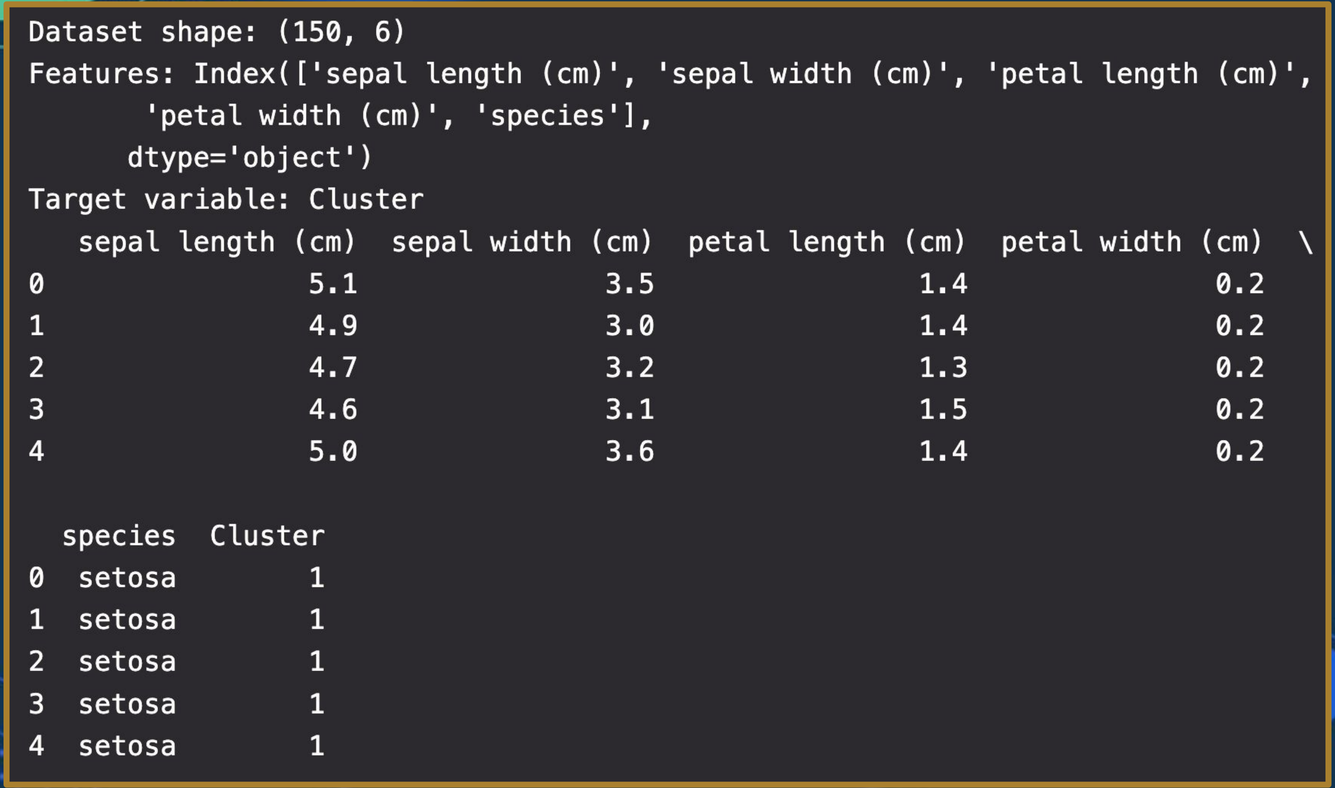
```python
# Load the Iris dataset
iris = load_iris()
data = pd.DataFrame(data=iris.data, columns=iris.feature_names)
data['species'] = iris.target_names[iris.target]
```

CoGrammar

Loading and Exploring the Dataset

- ❖ After loading the dataset, we'll explore its basic properties:
  - ➢ **Shape of the dataset:** number of rows and columns
  - ➢ **Features:** the independent variables in the dataset
  - ➢ **Target variable:** the dependent variable (depends on the features) we want to predict or analyse

CoGrammar



```
Dataset shape: (150, 6)
Features: Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'species'],
      dtype='object')
Target variable: Cluster
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

  species  Cluster
0  setosa        1
1  setosa        1
2  setosa        1
3  setosa        1
4  setosa        1
```

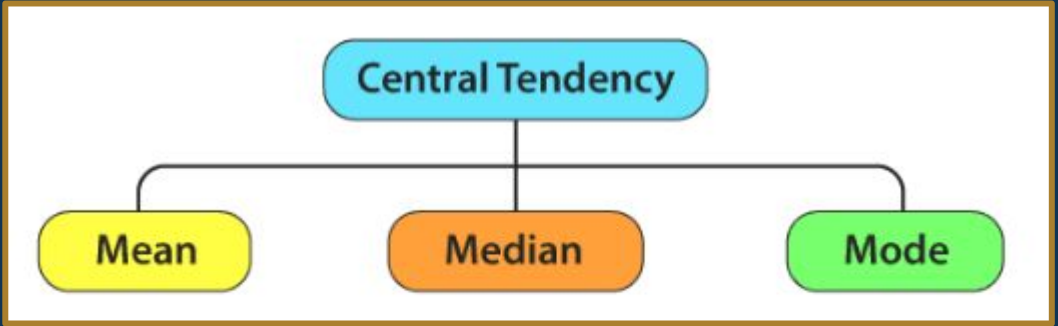
CoGrammar

Univariate Analysis

- ❖ Univariate analysis involves analysing each variable individually.
    - ➢ **Univariate:** involving one variate or variable quantity.
- ❖ We'll start by calculating descriptive statistics for the numeric variables using the describe() function.
- ❖ Descriptive statistics provide a summary of the central tendency, dispersion, and shape of the data.

CoGrammar

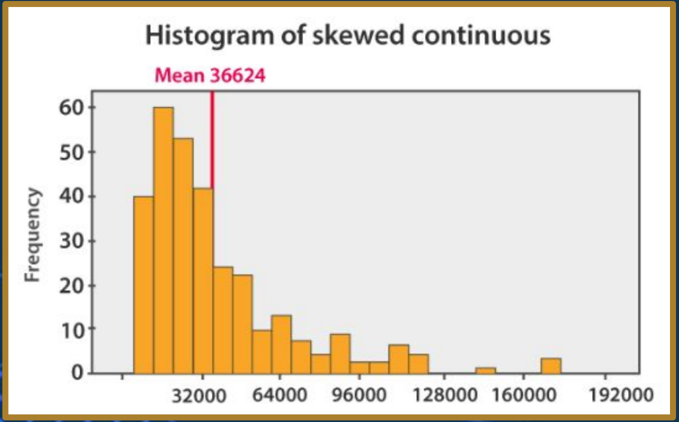
Measures of Central Tendency

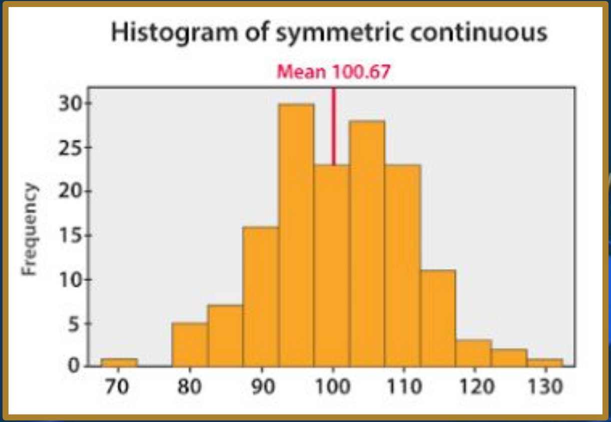


CoGrammar

Measures of Central Tendency

- **Mean** represents the average value of the dataset. It can be calculated as the sum of all the values in the dataset divided by the number of values.

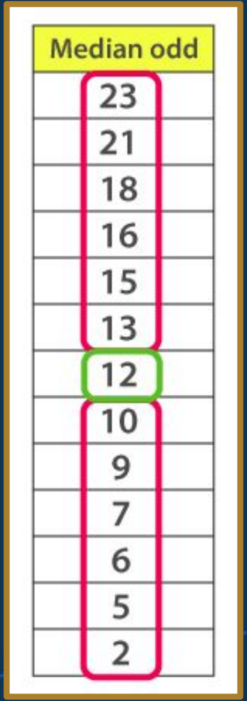




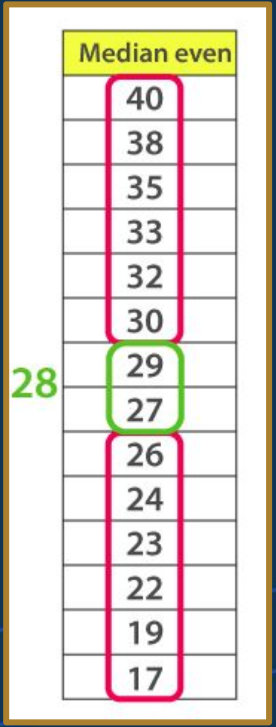
CoGrammar

Measures of Central Tendency

❖ **Median** is the middle value of the dataset in which the dataset is arranged in the ascending order or in descending order. When the dataset contains an even number of values, then the median value of the dataset can be found by taking the mean of the middle two values.



Median odd
23
21
18
16
15
13
12
10
9
7
6
5
2

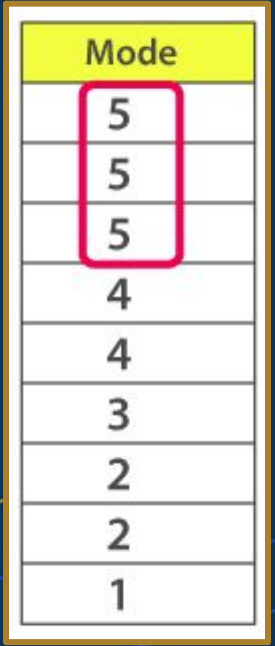


	Median even
	40
	38
	35
	33
	32
	30
28	29
	27
	26
	24
	23
	22
	19
	17

CoGrammar

Measures of Central Tendency

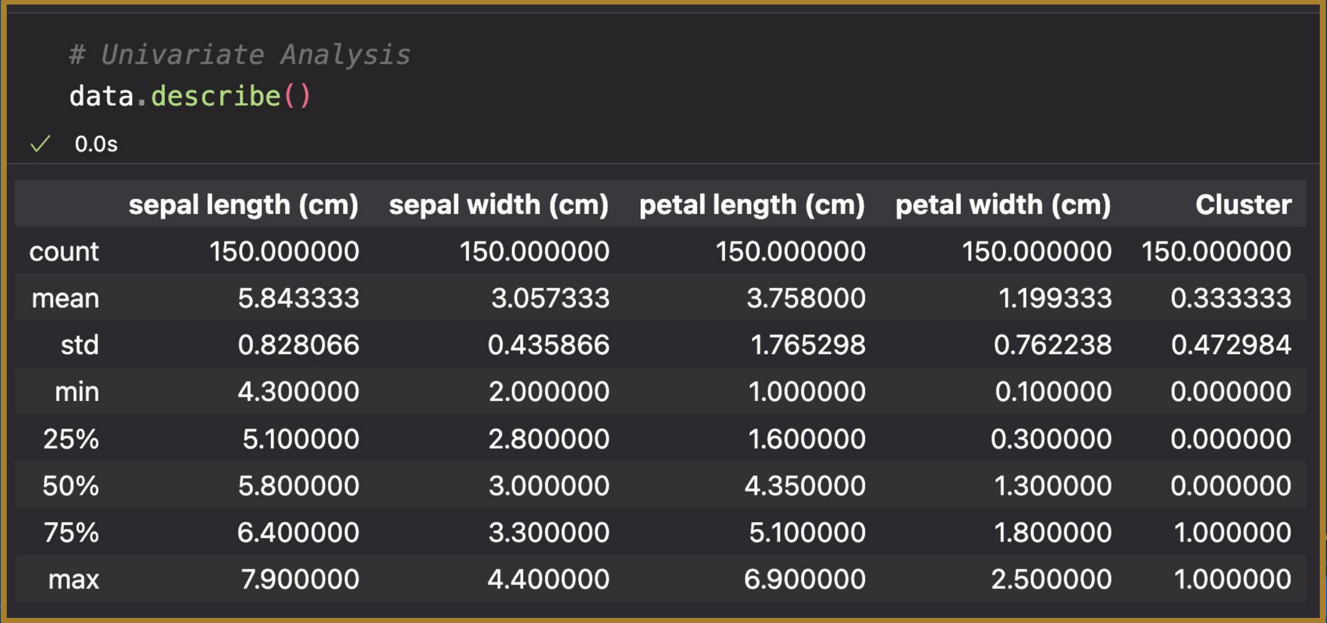
- **Mode** represents the frequently occurring value in the dataset. Sometimes the dataset may contain multiple modes and in some cases, it does not contain any mode at all.



Mode
5
5
5
4
4
3
2
2
1

CoGrammar

Univariate Analysis



```
# Univariate Analysis
data.describe()
```

✓ 0.0s

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Cluster
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	0.333333
std	0.828066	0.435866	1.765298	0.762238	0.472984
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	0.000000
75%	6.400000	3.300000	5.100000	1.800000	1.000000
max	7.900000	4.400000	6.900000	2.500000	1.000000

CoGrammar

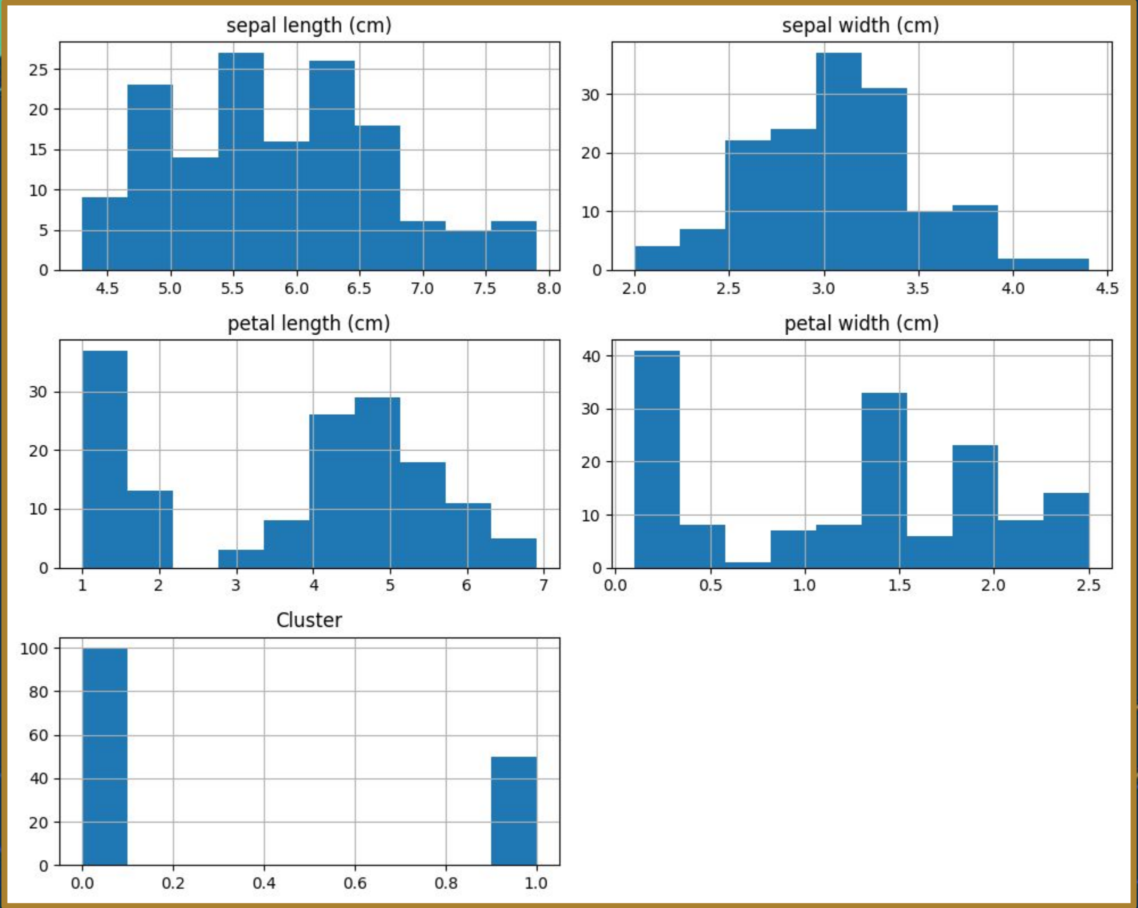
Univariate Analysis

❖ Next, we'll visualise the distribution of each feature using histograms and box plots.

❖ **Histograms** show the frequency distribution of a variable, helping to identify the shape, central tendency, and spread of the data.

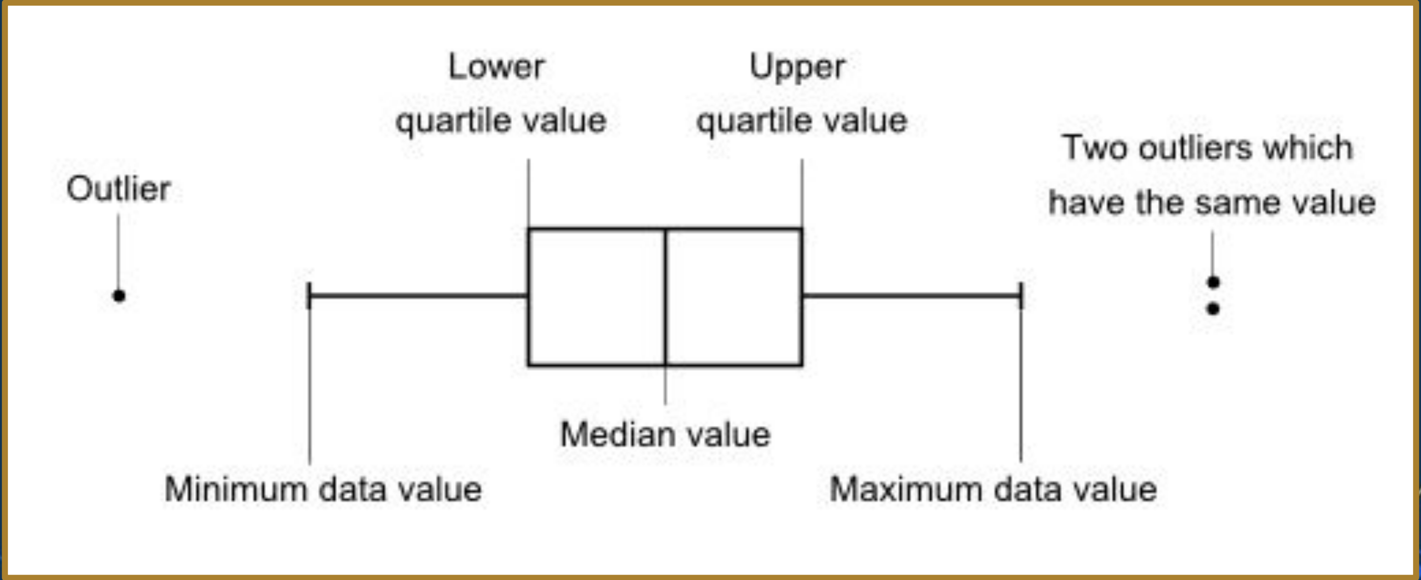
❖ **Box plots** provide a summary of the distribution, highlighting the median, quartiles, and outliers.

CoGrammar

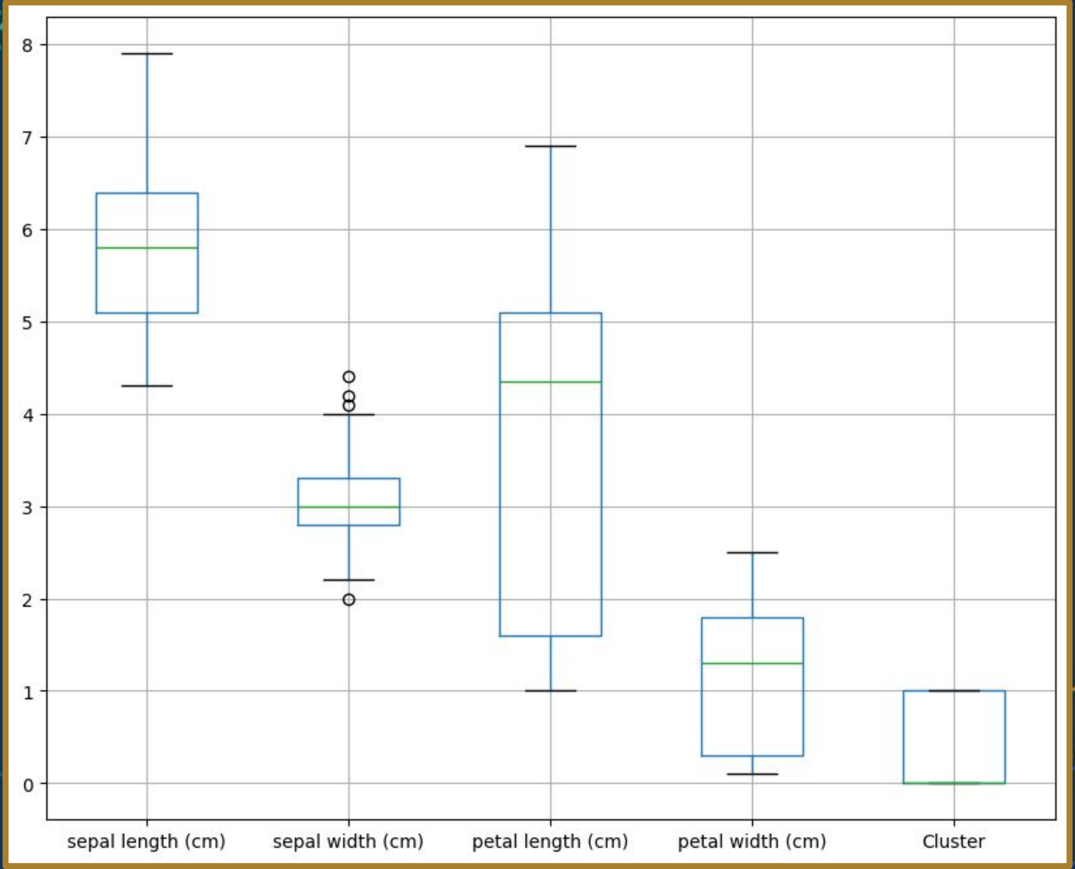

sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
Cluster

CoGrammar

Box Plots



CoGrammar

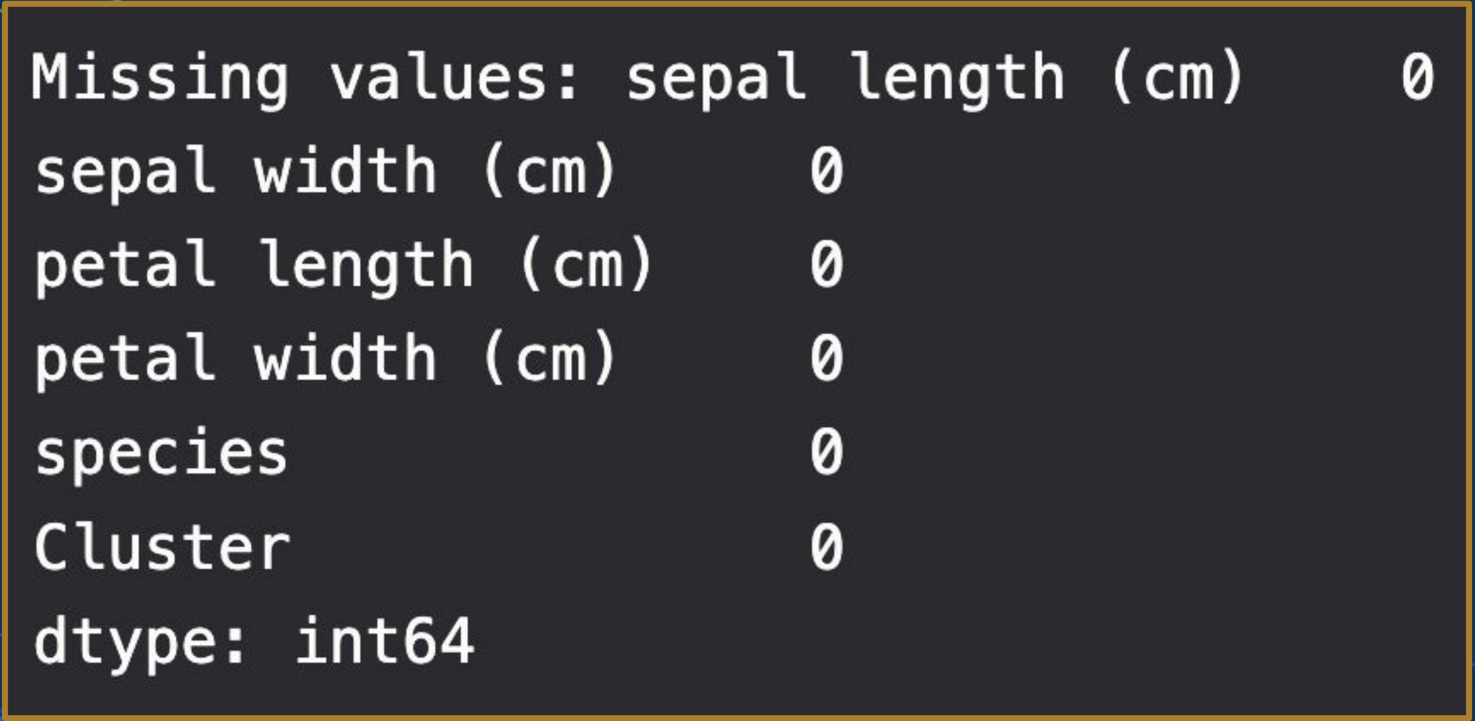

8
7
6
5
4
3
2
1
0
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
Cluster

CoGrammar

Univariate Analysis

- ❖ We'll also check for missing values in the dataset using the isnull().sum() function.
- ❖ Missing values can impact the analysis and need to be handled appropriately.
  - ➢ Common strategies include filling missing values with the mean, median, or mode.
  - ➢ It’s usually not a good idea to just drop data, as this could skew the data and thus the results of prediction.

CoGrammar

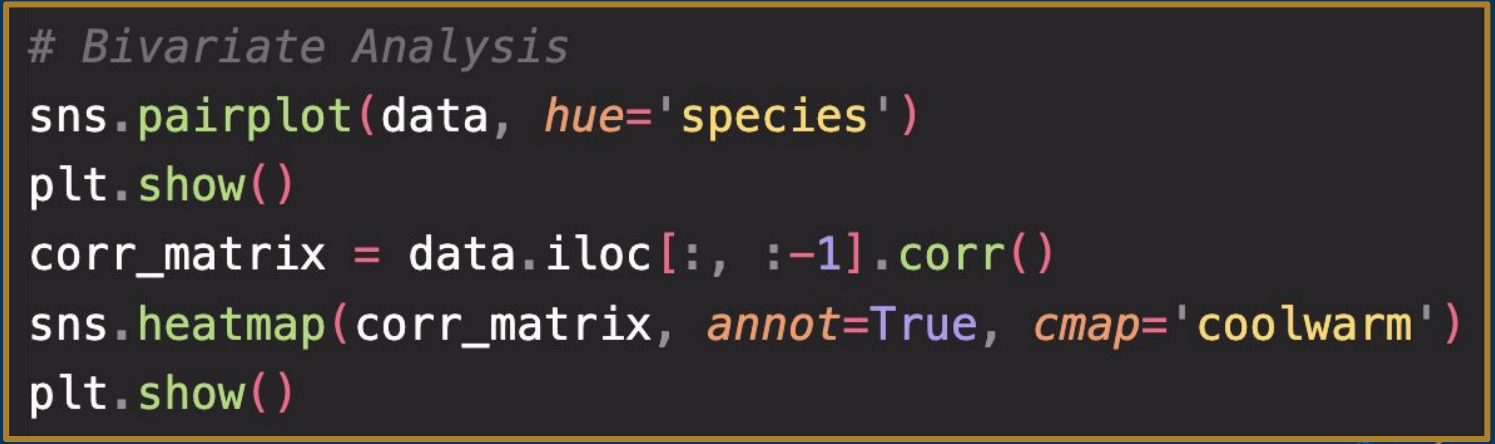


```
Missing values: sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
species              0
Cluster              0
dtype: int64
```

Bivariate Analysis

- ❖ Bivariate analysis involves examining the relationship between two variables.
    - ➢ **Bivariate:** involving or depending on two variates.
- ❖ We'll use scatter plots to visualise the relationship between features and the target variable.
- ❖ **Scatter plots** help identify patterns, correlations, and clusters in the data.

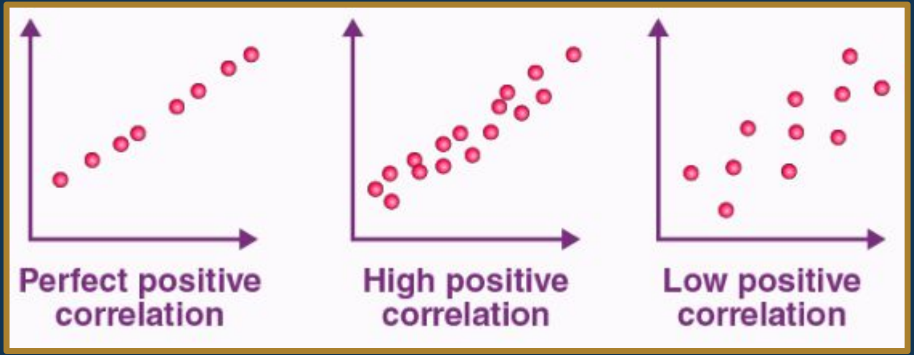
CoGrammar

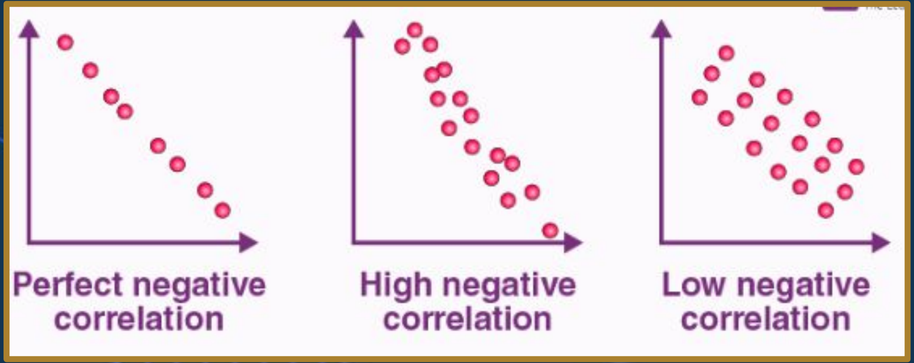


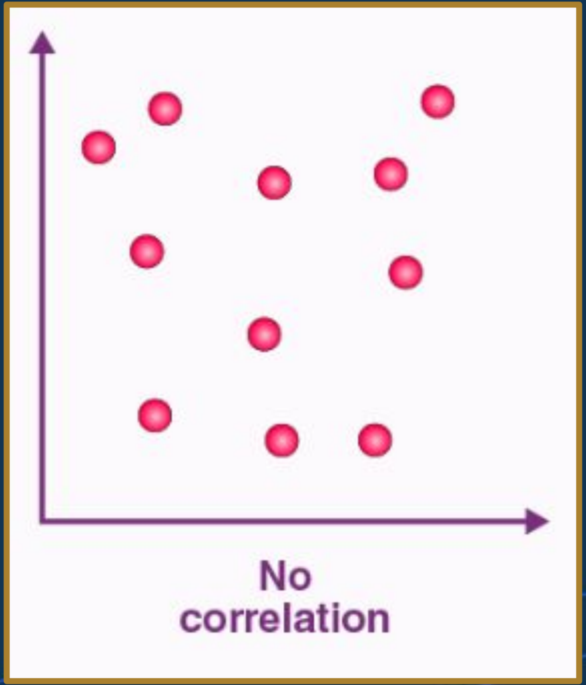
```python
# Bivariate Analysis
sns.pairplot(data, hue='species')
plt.show()
corr_matrix = data.iloc[:, :-1].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```

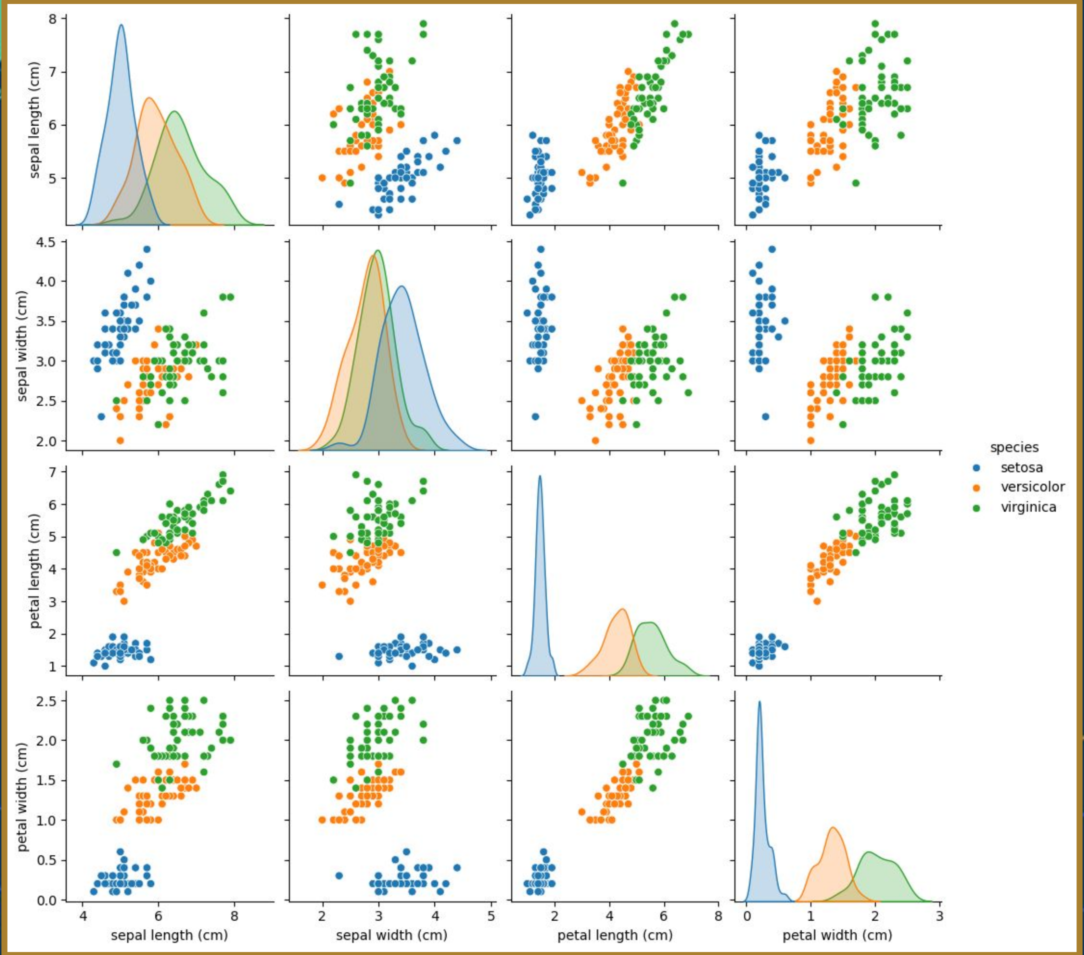
CoGrammar

Scatter Plot







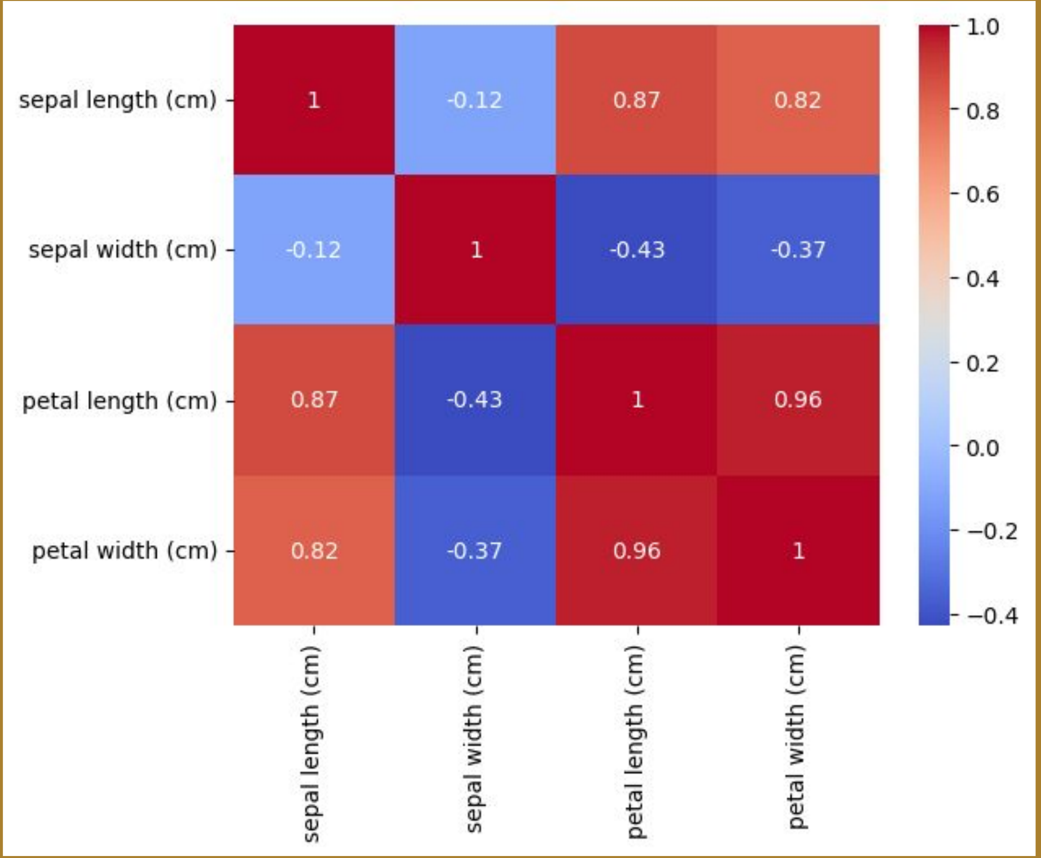

sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
species
setosa
versicolor
virginica

CoGrammar

Bivariate Analysis

- ❖ To quantify the relationship between numeric features, we'll calculate the correlation matrix.
- ❖ The correlation matrix shows the pairwise correlation coefficients between variables.
- ❖ We'll visualise the correlation matrix using a heatmap.

CoGrammar



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
sepal length (cm)	1	-0.12	0.87	0.82
sepal width (cm)	-0.12	1	-0.43	-0.37
petal length (cm)	0.87	-0.43	1	0.96
petal width (cm)	0.82	-0.37	0.96	1

Bivariate Analysis

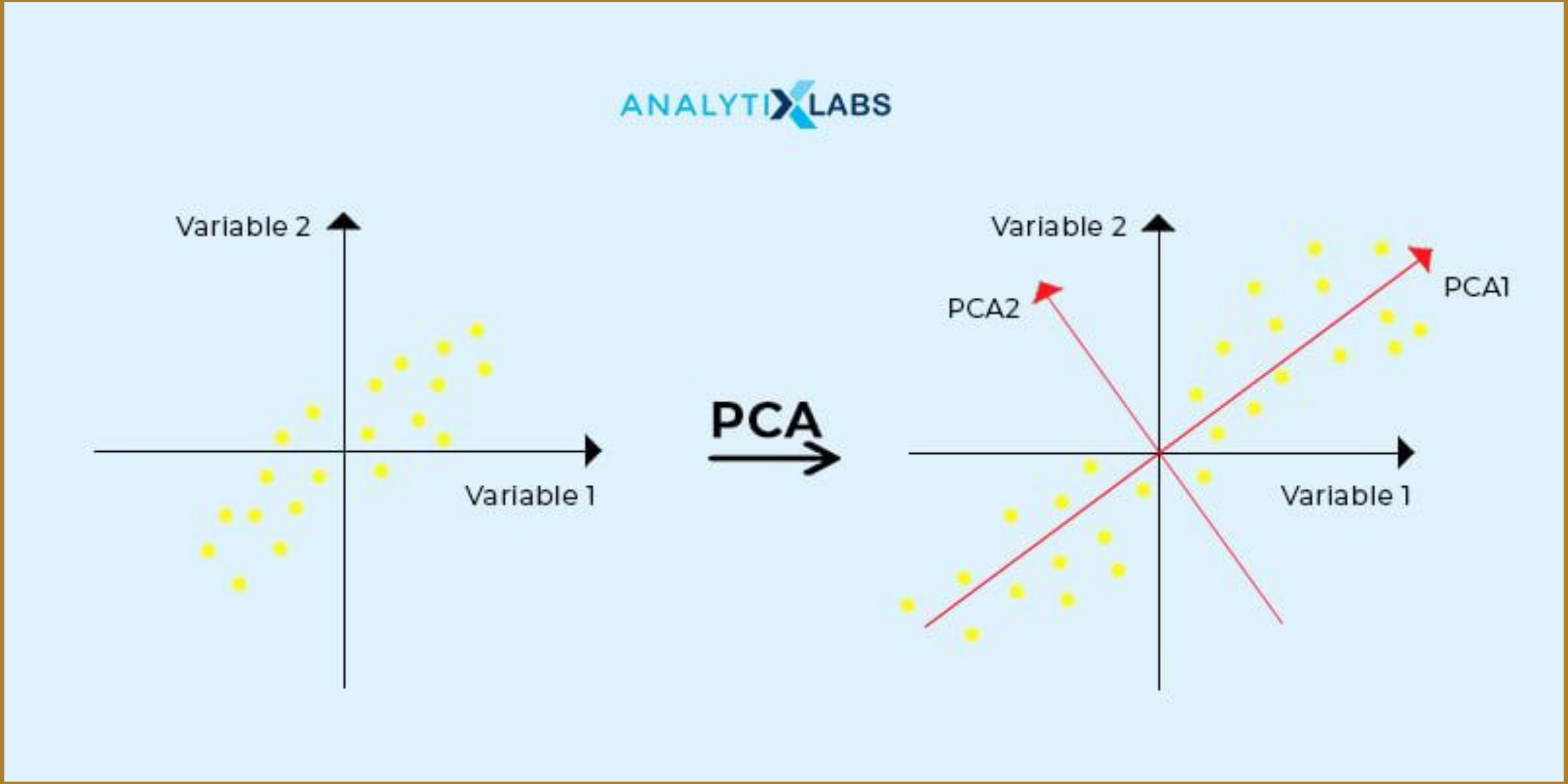
- Interpreting the correlation matrix:
  - Correlation coefficients range from -1 to 1.
  - The high positive correlations between Petal Length and Petal Width (0.96) and between Sepal Length and Petal Length (0.87) suggest that these pairs of features are strongly related and may provide similar information.
  - The low correlations between Sepal Width and the other features indicate that Sepal Width provides relatively independent information compared to the other features.

CoGrammar

Multivariate Analysis - PCA

- ❖ Multivariate analysis involves examining relationships among multiple variables simultaneously.
- ❖ **Principal Component Analysis (PCA)** is a dimensionality reduction technique that transforms the original features into a new set of uncorrelated features called principal components.
- ❖ PCA helps identify patterns and structure in high-dimensional data by finding the directions of maximum variance.

CoGrammar

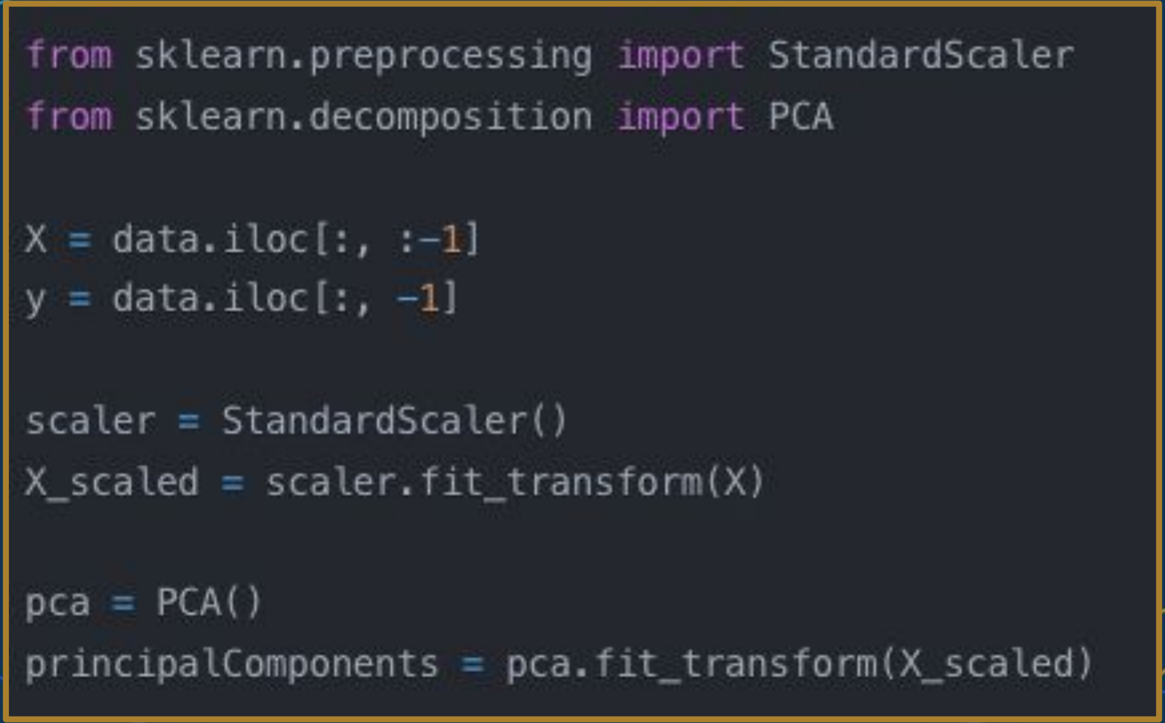


Source: [AnalytixLabs]

PCA Steps (Python abstracts all the math)

1. Standardize the data to ensure all features have zero mean and unit variance.
2. Compute the covariance matrix of the standardized data.
3. Calculate the eigenvectors and eigenvalues of the covariance matrix.
4. Sort the eigenvectors in descending order of their corresponding eigenvalues.
5. Select the top k eigenvectors as the principal components.
6. Transform the original data into the new feature space defined by the principal components.

CoGrammar

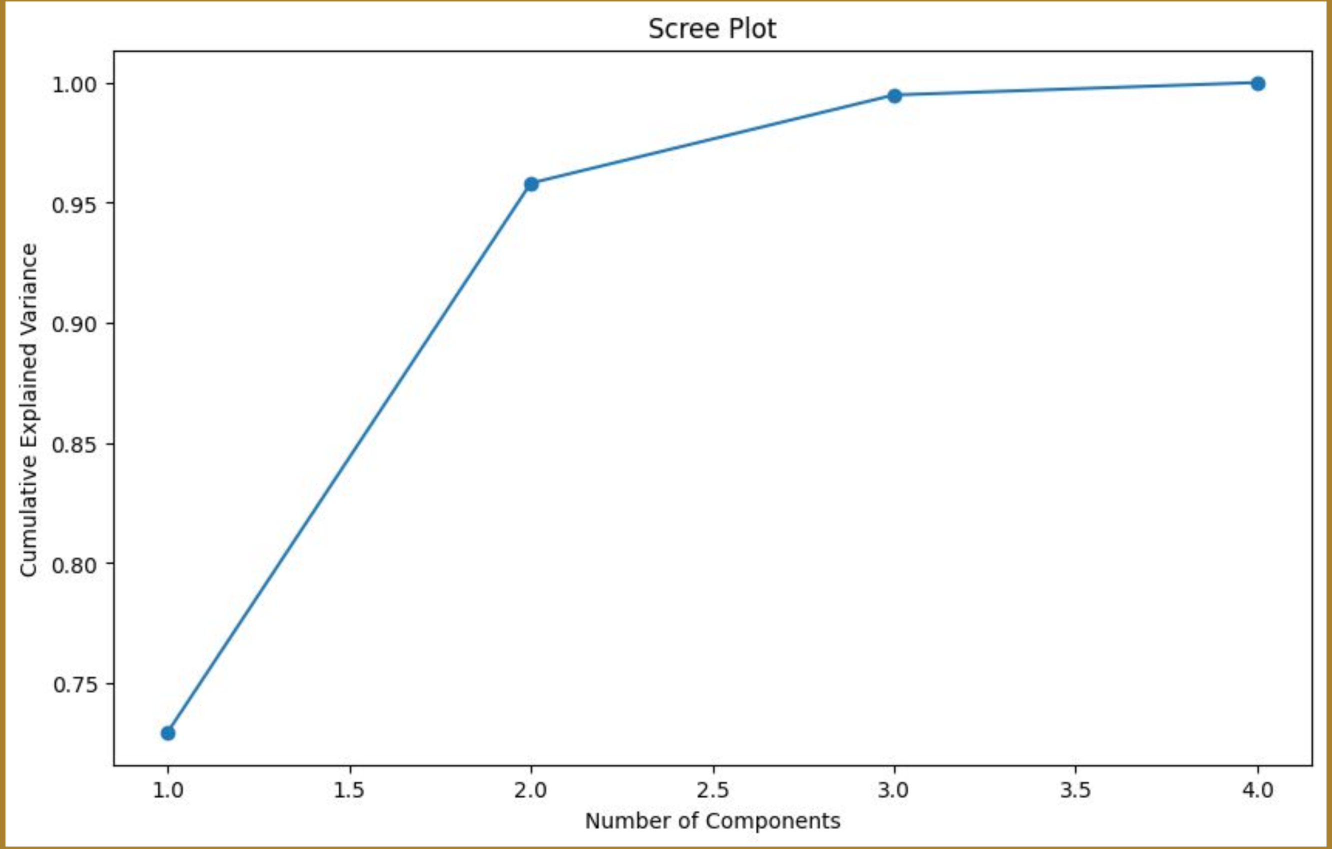

```python
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

X = data.iloc[:, :-1]
y = data.iloc[:, -1]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

pca = PCA()
principalComponents = pca.fit_transform(X_scaled)
```

CoGrammar

PCA

- ❖ To determine the number of principal components to retain, we can analyse the explained variance ratio.
- ❖ The explained variance ratio represents the proportion of variance explained by each principal component.
- ❖ We can visualise the cumulative explained variance using a scree plot.

CoGrammar



Scree Plot
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Number of Components

Scree Plot

- ❖ Interpreting the scree plot:
  - ➢ Look for an elbow point where the cumulative explained variance starts to plateau.
  - ➢ Choose the number of components that capture a significant portion of the total variance (e.g., 80-90%).
  - ➢ In our case it seems to be 2 components.

CoGrammar

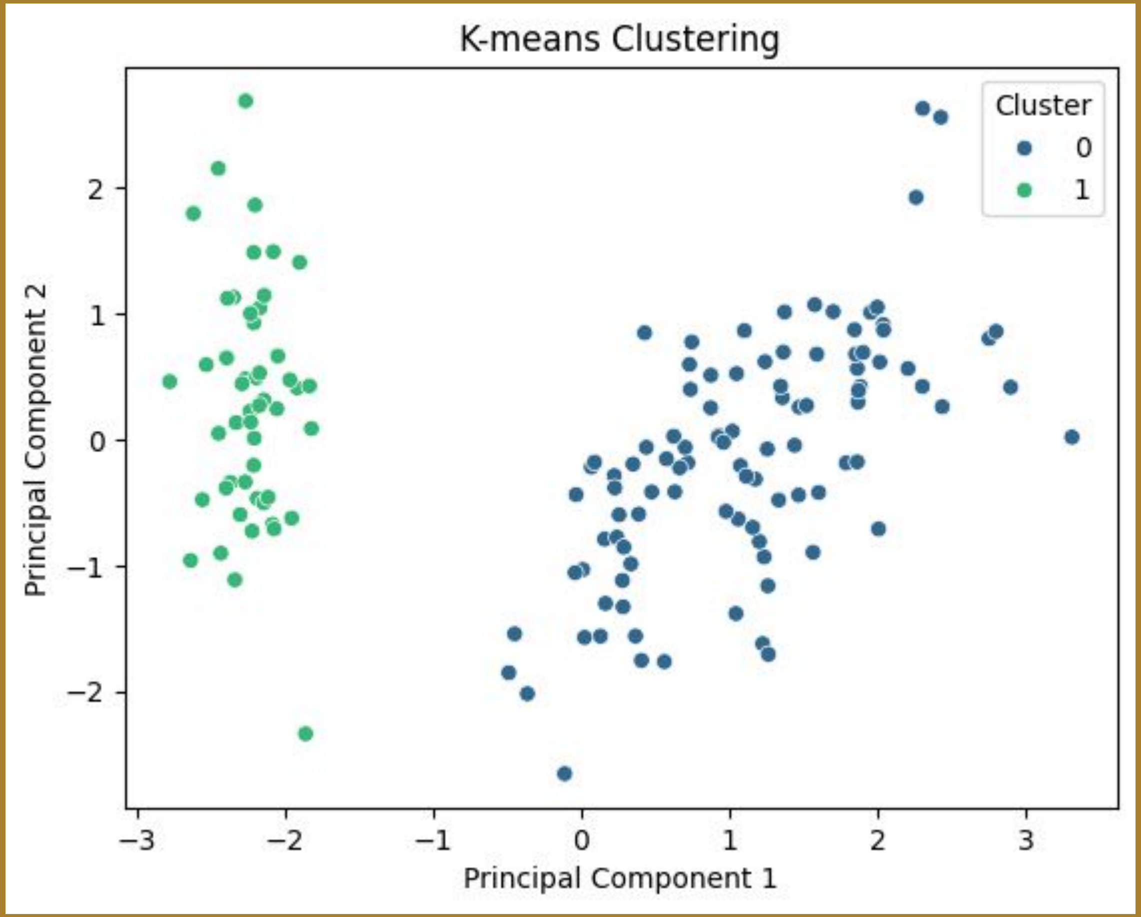
Multivariate Analysis - K-means Clustering

- K-means clustering is an unsupervised learning algorithm that partitions the data into K clusters based on similarity.
- It aims to minimize the within-cluster sum of squares (WCSS) or the Euclidean distance between data points and their cluster centroids.
- **More about this in later lectures.**

CoGrammar

K-means Clustering Steps (Python abstracts the math)

1. Choose the number of clusters K.
2. Initialize K cluster centroids randomly.
3. Assign each data point to the nearest centroid based on Euclidean distance.
4. Update the cluster centroids by taking the mean of the data points assigned to each cluster.
5. Repeat steps 3 and 4 until convergence or a maximum number of iterations is reached.


K-means Clustering
Principal Component 2
Principal Component 1
Cluster
0
1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

CoGrammar

Multivariate Analysis - K-means Clustering

- Interpreting the clustering results:
  - Observe the separation and compactness of the clusters.
  - Analyse the characteristics of data points within each cluster.
  - Consider the domain knowledge and interpret the meaning of the clusters.
    - In our case it seems like 2 plant species are very alike and one is distinctly different (setosa).

CoGrammar

Feature Importance

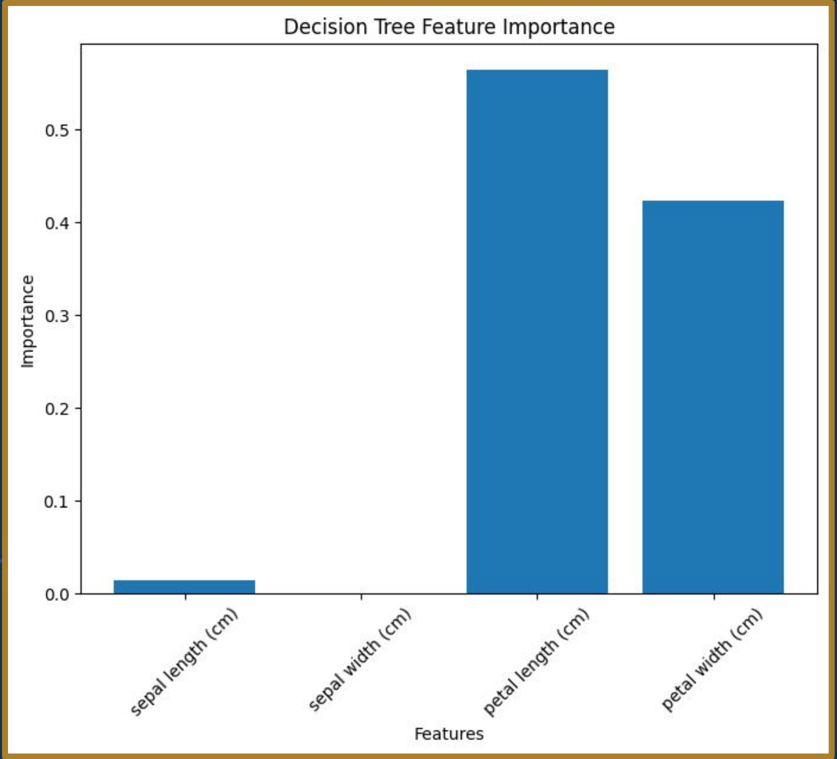
- ❖ Feature importance refers to the relative contribution of each feature in predicting the target variable.
- ❖ The prediction of a model is only as good as the features used to make the prediction, thus we want the most important predictors.
- ❖ We'll assess feature importance using statistical tests and machine learning techniques.

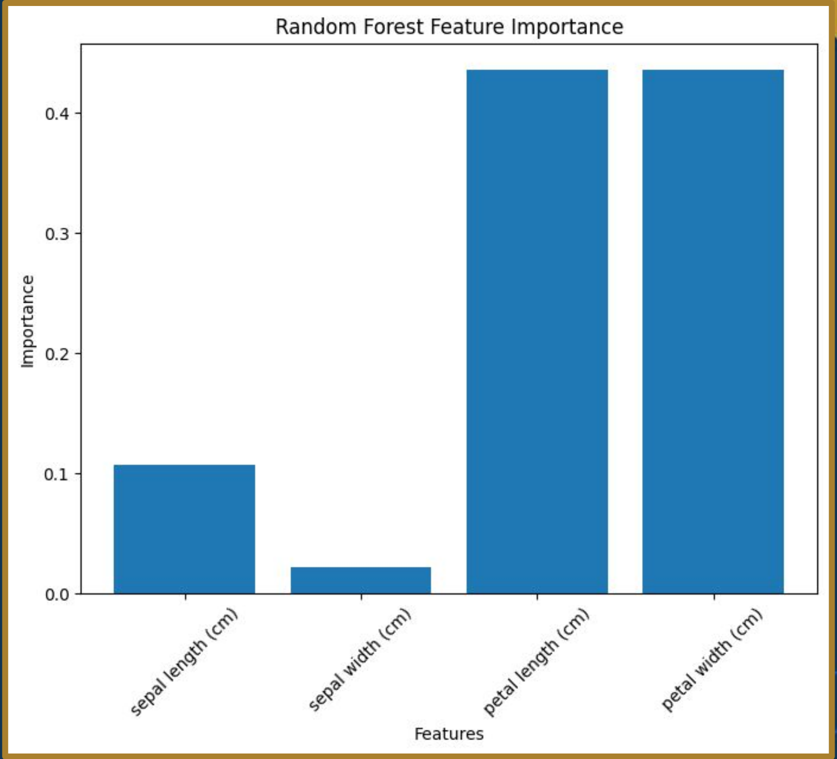
CoGrammar

Decision Trees and Random Forests

- Decision Trees and Random Forests are machine learning algorithms that can provide feature importance scores.
- The importance score represents the decrease in impurity or increase in information gain achieved by splitting on a particular feature.
- **More on these in dedicated lectures later on.**

CoGrammar


Decision Tree Feature Importance
Importance
0.0
0.1
0.2
0.3
0.4
0.5
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
Features


Random Forest Feature Importance
Importance
0.0
0.1
0.2
0.3
0.4
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
Features

CoGrammar

Questions and Answers

HyperionDev

Thank you for attending

HyperionDev