PYTHON PROJECT REPORT

The project is based on python programming language, where I have taken a data set to perform data cleaning, sorting, data transformation using pandas, statistical functions using numpy and visualization matplot and seaborn.

The data taken here is iris, below is some information on iris database:-

Overview of the Iris Dataset

- **Introduced by: Ronald A. Fisher in 1936**
- **Purpose: Originally used for linear discriminant analysis**
- **Dataset Size: 150 rows (samples), 5 columns (features + target)**
- **Classes (Species): 3 types of iris flowers**
  - ***Iris setosa***
  - ***Iris versicolor***
  - ***Iris virginica***

Features of the Dataset

Column Name	Description	Unit
sepal_length	Length of the sepal	centimeters
sepal_width	Width of the sepal	centimeters
petal_length	Length of the petal	centimeters

Column Name	**Description**	**Unit**
petal_width	**Width of the petal**	**centimeters**
species	**Type of Iris flower (target)**	**categorical (string)**

Why It’s Important

- **Ideal for classification problems**
- **Frequently used to teach basic data analysis and machine learning techniques**
- **Simple, clean, and well-balanced dataset (50 samples per class)**

What You Can Do with It

- **Data visualization (e.g., scatter plots, pair plots, histograms)**
- **Statistical summaries (mean, median, mode)**
- **Classification using:**
  - **k-NN**
  - **Decision Trees**
  - **SVM**
  - **Logistic Regression**
- **Dimensionality reduction (e.g., PCA)**
- **Correlation analysis**

Here's a detailed report on the Iris dataset and the Python code you've written for its analysis. This report explains what each part of the code does and how it contributes to understanding the dataset.

Dataset Overview: The Iris Dataset

The Iris dataset is a classic dataset in machine learning and statistics, widely used for pattern recognition and classification. It was introduced by British biologist and statistician Ronald Fisher in 1936.

Dataset Features:

- **Number of Samples: 150**
- **Number of Features: 4 numerical features**
- **Target/Label: Species (categorical)**

Feature Columns:

Feature	**Description**
sepal_length	**Length of the sepal (in cm)**
sepal_width	**Width of the sepal (in cm)**
petal_length	**Length of the petal (in cm)**
petal_width	**Width of the petal (in cm)**
species	**Type of Iris flower (setosa, versicolor, virginica)**

Explanation of the Code

1. Importing Libraries

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from scipy import stats
```

These libraries are essential for:

- **pandas: Data loading and manipulation**
- **numpy: Numerical operations**
- **seaborn: Statistical data visualization**
- **matplotlib.pyplot: Plotting**
- **scipy.stats: Advanced statistical functions (though not used in this code directly)**

---

2. Loading the Dataset

```
iris = sns.load_dataset('iris')
```

This line loads the Iris dataset from seaborn's built-in datasets. It returns a pandas DataFrame containing all 150 records.

---

3. Viewing the Dataset

```
print("Head of the dataset:")
```

```
print(iris.head())
```

This shows the first 5 rows of the dataset, giving a preview of the structure, types of data, and potential insights.

---

4. Null Values Check

```
print("\nNull values in the dataset:")
```

```
print(iris.isnull().sum())
```

Checks for missing values in all columns. This is important for data cleaning. In the Iris dataset, all values are complete — so the output will show zero nulls.

---

5. Statistical Calculations

```
mean_sepal_length = iris['sepal_length'].mean()
```

```
median_sepal_width = iris['sepal_width'].median()
```

```
mode_petal_length = iris['petal_length'].mode()[0]
```

These are basic statistical measures:

- Mean: Average of sepal length
- Median: Middle value of sepal width
- Mode: Most frequent value of petal length

Useful to understand the central tendency and distribution characteristics of each feature.

---

6. Correlation Heatmap

```
sns.heatmap(iris[['sepal_length', 'petal_length']].corr(), annot=True, cmap='coolwarm')
```

A heatmap displays the correlation coefficient between sepal length and petal length. Correlation values range from:

- **+1 = Strong positive correlation**
- **0 = No correlation**
- **–1 = Strong negative correlation**

Since these two features tend to increase together, the value is likely positive and relatively high (usually ~0.87).

Additional Visualizations

These visualizations give deeper insight into the distribution and relationships of features.

🔗 1. Pairplot

```
sns.pairplot(iris, hue='species')
```

Creates a grid of plots showing relationships between all pairs of features, colored by species. Ideal for spotting clusters and separability between species.

2. Boxplot

```
sns.boxplot(x='species', y='sepal_width', data=iris)
```

Shows distribution, median, and outliers of sepal width for each species using a box-and-whisker plot.

3. Violin Plot

```
sns.violinplot(x='species', y='petal_length', data=iris)
```

Like a boxplot, but also shows the kernel density estimate, helping visualize the distribution shape of petal length across species.

4. Histogram

```
plt.hist(iris['sepal_length'], bins=10, color='skyblue', edgecolor='black')
```

Displays the frequency distribution of sepal lengths. Useful to understand how the values are spread and whether the data is skewed or normal.

Summary of Findings

Analysis Type	Insights
Data Quality	No null values; clean dataset
Central Tendency	Mean, median, mode calculated to get core feature values
Correlation	Strong correlation between sepal and petal length
Distribution	Histogram and boxplots show distributions and outliers
Species Differences	Pairplots and violin plots show species-based separation
Visualization Techniques	Comprehensive visual insights using seaborn and matplotlib

---

- ## Conclusion

This code provides a solid exploratory data analysis (EDA) pipeline for the Iris dataset. It checks data quality, computes basic statistics, and uses effective visualizations to draw meaningful insights. This kind of EDA is a first and crucial step before building any machine learning models, as it helps you understand feature relationships, variances, and species classification potential.

Let me know if you want a version of this as a PDF report or notebook!