□ Project Report: Iris Flower Dataset Analysis and Predictions

Intern Name: Chukka Ganesh
Internship Title: Tech Internship at Cloudcredits
Internship Duration: April 4, 2025 – June 4, 2025
Project Title: Iris Flower Dataset Analysis and Predictions

Acknowledgment

I sincerely thank Cloudcredits for providing me this internship opportunity. I also extend my gratitude to mentors and peers for their valuable support and guidance throughout this learning journey.

Table of Contents

1. Introduction

The Iris flower dataset is one of the most famous databases in pattern recognition. This project aims to classify the species of Iris flowers based on petal and sepal dimensions using the K-Nearest Neighbors (KNN) algorithm.

2. Objective

To develop a classification model using KNN that accurately predicts the species of an Iris flower based on input features like sepal length, sepal width, petal length, and petal width.

3. Technology Stack

- **Programming Language**: Python
- **Development Tools**: Jupyter Notebook
- **Libraries Used**:
  - numpy
  - pandas
  - matplotlib
  - seaborn
  - scikit-learn

---

4. Methodology

1. **Data Collection**:
   Loaded the Iris dataset using Seaborn’s built-in dataset repository.
2. **Data Preprocessing**:
   Checked for null values, visualized the dataset, and encoded categorical values if needed.
3. **Model Building**:
   Applied K-Nearest Neighbors (KNN) from `sklearn.neighbors`.
4. **Model Evaluation**:
   Used accuracy score, confusion matrix, and visual plots for performance evaluation.

---

5. Implementation Details

Libraries and Dataset:

```
python
CopyEdit
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

# Load dataset
data = sns.load_dataset("iris")
```

Data Preprocessing:

```
python
CopyEdit
# Split data
X = data.drop("species", axis=1)
```

```
y = data["species"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Model Training and Evaluation:

```
python
CopyEdit
# Train KNN model
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)

# Predict and Evaluate
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

---

6. Results and Evaluation

- The KNN classifier achieved an accuracy of approximately **96-98%** on the test set.
- The confusion matrix showed strong class-wise prediction with minimal misclassification.
- The model is efficient and performs well with this small dataset.

---

7. Conclusion

This project demonstrates the application of the KNN algorithm on the Iris dataset, achieving high accuracy. The simplicity and effectiveness of KNN make it an ideal choice for small-scale classification tasks.

---

8. References

- https://seaborn.pydata.org
- https://scikit-learn.org
- https://pandas.pydata.org
- https://matplotlib.org

9. Appendix

All code is available in the Jupyter notebook file named `IRIS.ipynb`.