{AUTUMN INTERNSHIP PROJECT REPORT FORMAT}

**Project Title :Classification with logistic regression and random forest for IRIS dataset**

Debanjan Samanta

BTech (CSE) RCC Institute of Information Technology

Period of Internship: 25th August 2025 - 19th September 2025 (Do not change the dates)

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

**1. Abstract**

This project focuses on applying machine learning techniques to classify the Iris species using the well-known Iris dataset. The dataset contains 150 samples of three Iris species (*Setosa, Versicolor, Virginica*) with four features: sepal length, sepal width, petal length, and petal width. The objective was to preprocess the dataset, perform exploratory data analysis (EDA), and apply classification models such as Logistic Regression and Random Forest. Accuracy, precision, recall, and F1-scores were used to evaluate performance. Logistic Regression achieved ~**93% accuracy**, while Random Forest achieved ~**88%**, showing that in this dataset, the linear model outperformed the ensemble model. The project demonstrates how supervised learning can effectively distinguish between Iris species and highlights the role of feature selection and visualization in classification tasks.

**2. Introduction**

The Iris dataset is a classical dataset in machine learning and statistics, widely used for classification problems. It contains measurements of sepal and petal lengths and widths for 150 Iris flowers across three species. The simplicity and clear structure make it ideal for demonstrating ML workflows.

This project adapts a standard ML pipeline: data loading, preprocessing, visualization, train-test split, model training, and evaluation. The methodology allows comparison of linear and ensemble classifiers.

**Relevance:**

* Introduces machine learning concepts through a well-understood dataset.
* Highlights the importance of feature selection and scaling.
* Demonstrates model evaluation and comparison techniques.

**Technology involved:**

* Python 3, Google Colab, scikit-learn, pandas, seaborn, matplotlib.

**Topics covered during internship training (first two weeks):**

1. Python programming and pandas for data handling.
2. Exploratory data analysis (EDA).
3. Data visualization with matplotlib and seaborn.
4. Machine learning basics: classification models.
5. Train-test splitting and evaluation metrics.
6. Logistic Regression and Random Forest.
7. GitHub usage for code management.

**3. Project Objectives**

* To understand and preprocess the Iris dataset for machine learning applications.
* To perform exploratory data analysis (EDA) and visualize relationships between features.
* To apply Logistic Regression and Random Forest for classification of Iris species.
* To evaluate and compare model performance using accuracy, confusion matrices, and classification reports.
* To demonstrate the impact of linear vs non-linear models in classification tasks.

**4. Methodology**

**Step 1: Data Collection**

* The Iris dataset is available in scikit-learn; it was loaded directly using load\_iris().

**Step 2: Data Preprocessing**

* Converted numeric target labels to species names (*Setosa, Versicolor, Virginica*).
* Combined feature and target data into a single DataFrame for visualization.

**Step 3: Exploratory Data Analysis (EDA)**

* Calculated summary statistics for all features.
* Plotted correlation heatmaps to identify strongly correlated features.
* Created pairplots to visualize separation between species.

**Step 4: Train-Test Split**

* Data was split into 70% training and 30% testing sets using stratified sampling to maintain species distribution.

**Step 5: Model Training**

* **Logistic Regression**: Linear model with max\_iter=200.
* **Random Forest**: Ensemble model with 100 decision trees (n\_estimators=100).

**Step 6: Model Evaluation**

* Accuracy, classification reports, and confusion matrices were generated.
* Visualizations were created using seaborn heatmaps.

**Step 7: Code Management**

* The Python notebook was managed in Google Colab.
* Codes were pushed to GitHub for version control.

**Flow of Work:**

Data Loading → Data Preprocessing → EDA → Train-Test Split → Model Training → Model Evaluation → Result Comparison

**5. Data Analysis and Results**

**5.1 Descriptive Analysis**

* Dataset contains 150 samples and 4 features.
* Each species has 50 samples.
* Petal length and petal width are highly correlated and most useful for separating species.

**5.2 Visualizations**

* **Pairplots** showed clear separation of *Setosa* and partial overlap of *Versicolor* and *Virginica*.
* **Heatmap** highlighted strong correlation between petal length and petal width.
* **Boxplots** confirmed distinct distributions for features across species.

**5.3 Model Results**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Logistic Regression | ~93% | 0.93 | 0.93 | 0.93 |
| Random Forest | ~88% | 0.88 | 0.88 | 0.88 |

**Confusion Matrices:**

* Logistic Regression misclassified a few *Versicolor* and *Virginica* samples.
* Random Forest misclassified more samples, likely due to overfitting on a small dataset with limited features.

**6. Conclusion**

The project demonstrates the effectiveness of machine learning in classifying Iris species. Logistic Regression performed better (~93%) than Random Forest (~88%) in this small, well-structured dataset, due to the dataset being nearly linearly separable. Petal length and width remain the most informative features. Random Forest may overfit on smaller datasets, explaining its slightly lower performance.

**Key Takeaways:**

* Logistic Regression is suitable for datasets with linear separability.
* Random Forest is more robust to complex patterns but can overfit small datasets.
* Feature selection and visualization are crucial for understanding the dataset.

**Recommendations for future work:**

* Explore hyperparameter tuning for Random Forest.
* Apply other classifiers such as SVM or Gradient Boosting.
* Use synthetic data to augment smaller classes or handle imbalance.

**7. Appendices**

**Appendix A: References**

1. Scikit-learn: Iris dataset documentation – https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load\_iris.html
2. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” JMLR, 2011.
3. James et al., *An Introduction to Statistical Learning*, Springer, 2013.