Sloan Digital Sky Survey (SDSS) Galaxy Classification using Machine Learning

Name: Muhammed Najeed P

Date: 05/10/2025

Submitted to

SmartInternz Internship Mentor

Contact: najeedmnp7860@gmail.com

Abstract

This project focuses on classifying galaxies using the Sloan Digital Sky Survey (SDSS) dataset. The main goal is to identify whether a galaxy belongs to the “Starforming” or “Starburst” subclass using machine learning models.

The project includes steps such as data collection, cleaning, exploratory data analysis (EDA), feature selection, data balancing using SMOTE, model training with multiple algorithms, and final deployment using a Flask web application.

Among the models used — Decision Tree, Logistic Regression, and Random Forest — the Random Forest classifier provided the best accuracy.

Project Workflow

1. **Data Collection & Preparation**
    - Collected the SDSS galaxy dataset.
    - Handled missing values and converted datatypes.
    - Encoded subclass labels for modeling.
2. **Exploratory Data Analysis**
    - Conducted descriptive and visual analyses.
    - Examined correlations and removed outliers using IQR method.
3. **Feature Selection & Balancing**
    - Selected top 10 features using *SelectKBest*.
    - Balanced classes using *SMOTE* (oversampling).
4. **Model Building**
    - Trained multiple algorithms: Decision Tree, Logistic Regression, and Random Forest.
    - Compared accuracy and performance metrics.
5. **Model Deployment**
    - Saved the best model.
    - Integrated with a Flask web framework.
    - Built simple HTML pages for user input and result display.

Phase 1: Data Collection & Preparation

Objectives: Collect a clean dataset suitable for training machine learning models.

Tasks:

1. **Collect the Dataset:** The SDSS dataset containing galaxy properties and subclass labels was obtained.
2. **Import Libraries:** Python libraries such as Pandas, Numpy, Matplotlib, and Seaborn were imported for data manipulation and visualization.
3. **Read Dataset:** The dataset was loaded into a dataframe for processing.
4. **Data Cleaning:** Unnecessary columns were removed, and missing values were handled.
5. **Data Transformation:** The subclass column (object type) was converted into integers using label encoding for machine learning compatibility.

Outcome: A clean dataset with numeric features and encoded target variable ready for analysis

Phase 2: Exploratory Data Analysis (EDA)

Objectives: Understand data distribution, detect patterns, and identify outliers.

Tasks:

1. **Descriptive Statistics:** Basic statistics (mean, median, standard deviation) were computed to summarize each feature.
2. **Visual Analysis:** Box plots, histograms, and correlation heatmaps were created to examine feature distributions and relationships.
3. **Univariate Analysis:** Each feature was analyzed individually using pie charts and histograms to understand its distribution.
4. **Bivariate Analysis:** Relationships between features and the target variable were explored using bar plots and scatter plots.

5. **Multivariate Analysis:** Feature interactions were visualized with correlation matrices.
6. **Handling Outliers:** Extreme values were capped using the interquartile range (IQR) method to reduce their impact.

Outcome: Insights into feature importance and data quality were obtained, preparing for model training.

Phase 3: Feature Selection & Data Balancing

Objectives: Reduce dimensionality, improve model performance, and address class imbalance.

Tasks:

1. **SelectKBest Algorithm:** Top 10 features were selected based on statistical scoring to enhance predictive power.
2. **SMOTE (Synthetic Minority Oversampling Technique):** Applied to balance the class distribution, ensuring the model does not bias toward the majority class.

Outcome: A balanced dataset with the most informative features ready for training robust machine learning models.

Phase 4: Model Building

Objectives: Train multiple machine learning models and identify the best-performing one.

Tasks:

1. **Splitting Data:** Dataset divided into training and testing sets to evaluate model performance.
2. **Scaling Features:** StandardScaler used to normalize feature values for improved model convergence.
3. **Training Models:** Decision Tree, Logistic Regression, and Random Forest models were trained.

4. **Model Evaluation:** Accuracy, confusion matrix, and classification reports were used to compare models.
5. **Selecting the Best Model:** Random Forest achieved the highest accuracy and stability.

Outcome: A high-performing machine learning model capable of predicting galaxy subclasses.

Phase 5: Model Deployment

Objectives: Deploy the trained model as a web application for user interaction.

Tasks:

1. **Save the Model:** Random Forest model was saved using joblib for future use without retraining.
2. **Build HTML Pages:** Created index.html for input and innerpage.html for displaying predictions.
3. **Build Python Code (Flask App):** Flask framework used to handle HTTP requests, retrieve inputs, pass them to the model, and display predictions.
4. **Run Web Application:** Tested locally on localhost, users can enter values and view predictions instantly.

Outcome: A functional web application integrated with the trained Random Forest model, ready for demonstration and further use.

Conclusion & Future Scope

Conclusion:
The project successfully automated the classification of galaxies using machine learning, achieving high accuracy with Random Forest and providing an interactive web application.

Future Scope:

- Implement deep learning for more accurate classification using images.

- Deploy the web app on cloud platforms for public access.
- Extend the model to handle additional galaxy subclasses and larger datasets.