Project - MLAI Internship

Predicting Income Level Using Classification Models

Context:

Income level prediction plays a crucial role in understanding socioeconomic factors and helping governments, researchers, and businesses design better policies, services, and targeted interventions. By predicting whether an individual's income exceeds $50K annually based on demographic and work-related attributes, machine learning models can assist in various decision-making processes, such as loan approvals, employment strategies, and policy analysis.

Objective:

The objective of this analysis is to build and compare various classification models to predict whether an individual's income is greater than $50,000 (Income = >50K) or not (Income = <=50K) based on demographic and employment details. The goal is to identify the best-performing model that can assist in informed decision-making regarding financial, social, and governmental initiatives.

Problem Statement:

Given the Adult Census Income Dataset, which contains demographic and employment-related data such as age, workclass, education, marital status, occupation, relationship, race, sex, capital gain, hours per week, and native country, the task is to apply different classification algorithms to predict the likelihood of an individual earning more than $50K per year.

The models to be applied include: Logistic Regression, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Naive Bayes, Decision Trees, Random Forests, MLP Classifier (Multi-layer Perceptron)

Link to Dataset:
https://github.com/tech4alltraining/aiml/raw/refs/heads/main/datasets/classification/adult.csv

NB:

#A few tasks (marked with *) weren't covered in class and are left for you to figure out and solve on your own. These tasks are optional for your project, but if you solve them, you'll get a better score!

Weekly Submission Plan

Week 1: Data Understanding, EDA, and Preprocessing

Tasks:

- Understand the data and problem statement clearly.
- Identify independent variables (features) and dependent variables (target = Income).
- Exploratory Data Analysis (EDA):
  - Determine the number of rows and columns in the dataset.
  - Show the data types of each column.
  - Generate summary statistics using the describe() method.
  - Identify and list the numeric and categorical features within the dataset.
  - Display the distinct values for each categorical feature.
  - Visualize bar graphs for three categorical features, illustrating the distribution of samples across each category*. [Hint: Use Matplotlib or Seaborn.]
  - Plot the correlation matrix (using heatmap) to explore relationships between numerical features*. [Hint: Use Matplotlib or Seaborn or DataFrame's corr() method.]

    - Examine the distribution of at least three numerical features*. [Hint: Use the DataFrame's hist() method.]
- Apply data preprocessing:
    - Handle missing values (represented as `?` in the dataset). [Hint: Convert all values represented as `?` into `NaN`, then proceed with handling missing values.]
    - For categorical columns, fill null values with the most frequent category (mode) within each column*.
    - Drop unwanted columns (if applicable).
    - Remove duplicated rows (if applicable).
    - Detect and handle outliers (if applicable).
    - Perform Label Encoding for categorical features (if applicable)..
    - Apply feature scaling where necessary.

Deliverable:

- Colab Notebook (.ipynb file) with EDA and preprocessing steps clearly documented.

Week 2: Model Building and Training

Tasks:

- Split the data into training and testing sets (e.g., 80:20 split) and Show sample size(s) of each split.
- Train the following classification models:
  Logistic Regression, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Naive Bayes, Decision Tree, Random Forest, MLP Classifier
- Learn, Understand and Train the two additional classification models*:
  GradientBoostingClassifier, XGBClassifier
- Make predictions on the testing data using each trained model.

Deliverable:

- Colab Notebook (.ipynb file) showing model training and prediction code.

Week 3: Model Evaluation, Comparison, and Conclusion

Tasks:

- Evaluate the performance of each model:
    - Accuracy Score
    - Classification Report (Accuracy, Precision, Recall, F1-Score),
    - Confusion Matrix (with visualization)* [Hint: Use matplotlib or seaborn]
- Compare the classifiers based on performance metrics.
- Write a final conclusion:
    - Which model performed best and why?
    - Any specific observations about the data or models?

Deliverable:

- Final Colab Notebook (.ipynb) summarizing evaluation results and conclusions.