Income Prediction Using Machine Learning and GUI Deployment

Bhuvanesh Sunil Nehete
EE23BTECH11043

Abstract

This paper presents a machine learning-based system for predicting whether an individual's income exceeds $50,000 per year using the Adult Census Income dataset. The project implements extensive preprocessing, label encoding, dataset balancing, model training, and evaluation using four classification algorithms. A Tkinter-based GUI is developed for real-time user interaction, enabling dynamic prediction based on demographic and occupational inputs. The paper describes the full end-to-end workflow with emphasis on dataset preparation and model training.

1 Introduction

Income prediction is a key socio-economic task with applications in policy design, market analysis, and employment assessment. The Adult Census Income dataset serves as a benchmark for binary classification. The motivation behind this project is to develop an end-to-end machine learning pipeline—from raw data to deployed GUI—that can provide real-time predictions.

2 Materials and Methods

2.1 Dataset Description

The Adult Census Income dataset contains demographic and employment details of individuals, including age, education, occupation, workclass, and hours-per-week. The goal is to predict whether a person's income exceeds $50K per year. It includes both categorical and numerical features and requires preprocessing due to missing values and class imbalance.

2.2 Data Preprocessing

Preprocessing included:

- Cleaning missing values
- Removing irrelevant columns
- Filtering noisy categories

- Label encoding categorical features
- Balancing dataset via undersampling

2.3 System Architecture

Figure 1 illustrates the workflow.

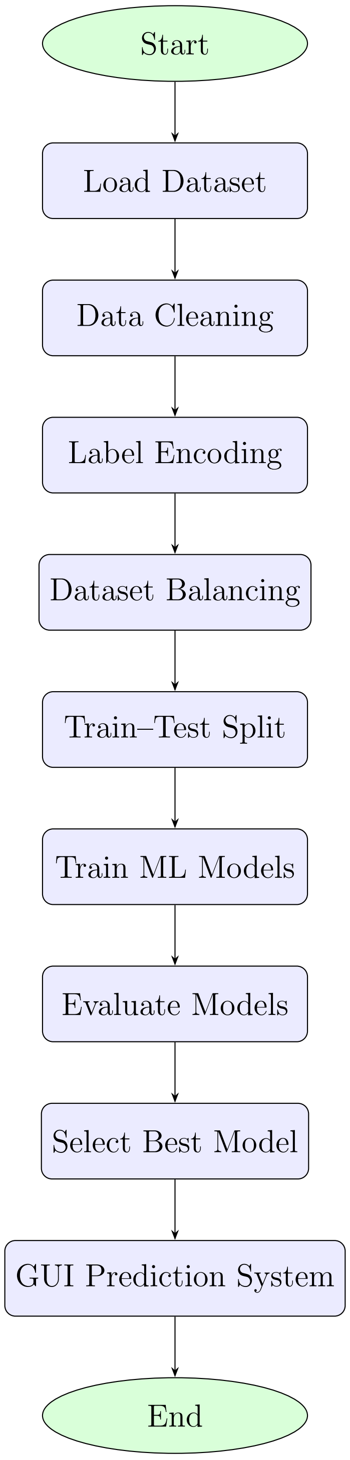

Figure 1: System Workflow Diagram

3 Training Process

3.1 Data Loading and Cleaning

The dataset was loaded using Pandas. Trailing spaces in column names were removed. Missing values represented as "?" were replaced with NaN and dropped. Several columns lacking predictive power were removed. Individuals with very low educational attainment were excluded to reduce noise and improve training stability.

3.2 Categorical Encoding

Several features such as workclass, occupation, gender, and marital-status are categorical. These were converted to numerical form using **Label Encoding**. For each categorical feature, a separate encoder was stored so that the GUI could encode new user inputs consistently.

3.3 Dataset Balancing

The Adult dataset is imbalanced. To avoid biased learning, the majority class (income = 0) was undersampled to match the size of the minority class (income = 1). The balanced dataset was then shuffled to ensure randomness.

3.4 Train–Test Split

The dataset was divided into training and testing subsets:

$$\text{Training Set} = 80\%, \quad \text{Testing Set} = 20\%$$

The testing data was kept isolated for unbiased evaluation.

3.5 Model Selection and Training

Four supervised classifiers were trained:

3.5.1 Logistic Regression

A linear model predicting probabilities using a sigmoid function. It is simple and effective when relationships are roughly linear. The model was trained using:

$$\text{Logistic Regression}(\text{max\_iter} = 1000)$$

3.5.2 Random Forest Classifier

An ensemble of decision trees trained using bootstrap aggregation. Trees learn on different subsets of data and features, reducing overfitting. The final prediction is obtained through majority voting.

3.5.3 Gradient Boosting Classifier

A boosting algorithm where trees are added sequentially. Each new tree corrects the errors of previous trees through gradient descent. Known for achieving high accuracy on structured datasets.

3.5.4 Support Vector Machine (SVM)

A margin-based classifier that finds the optimal separating hyperplane between classes. The default RBF kernel allows modeling nonlinear relationships.

3.6 Evaluation Metrics

After prediction on the test set, the following metrics were computed:

3.6.1 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

3.6.2 Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

3.6.3 Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

3.6.4 F1 Score

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

These metrics give a thorough understanding of model performance, especially since precision and recall must be balanced in classification tasks involving imbalanced data.

3.7 Best Model Selection

The model with the highest accuracy was chosen as the final model for GUI deployment:

$$\text{Best Model} = \underset{m \in \text{models}}{\text{argmax}}(\text{Accuracy}(m))$$

4 Results

4.1 Performance

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	75.28%	74.51%	77.50%	75.97%
Random Forest	78.62%	78.18%	79.91%	79.04%
Gradient Boosting	**82.45%**	79.91%	87.09%	83.35%
SVM	75.89%	73.37%	81.92%	77.41%

Table 1: Model Performance Metrics

The Gradient Boosting model achieved the best accuracy among all classifiers, showing strong overall performance. This confirms that ensemble methods work effectively on the income prediction task.

4.2 GUI Implementation

A Tkinter GUI was developed to allow real-time input of:

- Age
- Gender
- Occupation
- Workclass
- Hours per week
- Education level

User inputs are encoded using the stored label encoders. A single-row DataFrame is created and passed to the trained best model for prediction. The result is displayed as either $<= 50K$ or $> 50K$.

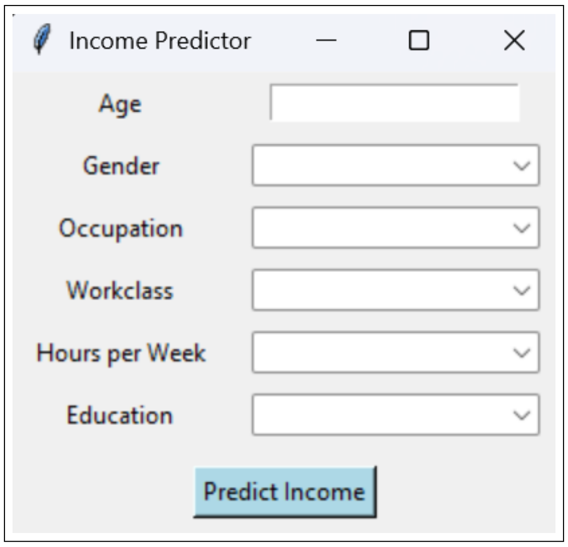

Figure 2: Graphical User Interface for Income Prediction.

5 Discussion

Important learnings include:

- Importance of preprocessing in improving model accuracy.
- Gradient Boosting outperforming classical models.
- Challenges in ensuring data encoding consistency for GUI deployment.

6 Conclusion

This project successfully demonstrates an end-to-end machine learning system for income prediction. With robust preprocessing, multiple model evaluations, and a user-friendly GUI, the system is suitable for learning, experimentation, and practical demonstration. Future improvements may include deploying the system on the web, integrating deep learning techniques, or expanding the dataset.