Census Income Prediction: A Machine Learning Approach

Sameer Singh Godara
sameer22439@iiitd.ac.in

Sanyam Barwar
sanyam22447@iiitd.ac.in

Sanyam Garg
sanyam22448@iiitd.ac.in

Vivan Rangra
vivan22581@iiitd.ac.in

Supervisor: Dr. Jainendra Shukla
Indraprastha Institute of Information Technology, Delhi
jainendra@iiitd.ac.in

Abstract

Predicting income levels based on demographic and socioeconomic factors is essential for various applications like policy-making, targeted marketing, and financial planning. This project focuses on comparing multiple machine learning models—such as Logistic Regression, Decision Trees, and Random Forest—to identify the most effective approach for predicting whether an individual's income is above $50,000. The Census Income dataset is used to train and evaluate the models. Initial results indicate that Random Forest provides the highest accuracy, though further fine-tuning and testing of models like Support Vector Machines and Neural Networks are required. For more details, please visit our GitHub repository: ML Project: Census Income Prediction.

1. Introduction

Income prediction is a crucial problem with practical implications for governments and businesses. Predicting whether an individual's income exceeds $50,000 based on features like age, education, occupation, and marital status can aid in better decision-making for policy development, targeted financial services, and marketing strategies. In this project, we tackle this problem using machine learning algorithms to assess which model offers the best trade-off between accuracy and computational efficiency.

Binary classification is employed to predict two classes: income greater than $50,000 or income less than or equal to $50,000. Our motivation stems from the growing demand for data-driven solutions in the socio-economic sector. The project also serves as a comparative study of several machine learning algorithms, aiming to find the most reliable model for income classification.

1.1. Problem Statement

Income prediction based on demographic data is a complex task due to the multiple factors influencing income levels. Our objective is to classify individuals into one of two categories: those earning more than $50,000 and those earning less. We aim to determine which machine learning model is best suited for this binary classification task in terms of accuracy, interpretability, and computational cost.

2. Literature Survey

A comprehensive literature review was conducted to understand existing methods in income classification using machine learning algorithms. The two most influential studies that guided our approach are outlined below:

- **Paper 1: Census Income Prediction using Machine Learning Techniques.** This paper applies multiple machine learning models—such as Logistic Regression, Decision Trees, and Random Forest—on the UCI Adult Income dataset. It highlights that Random Forest outperforms other models in terms of accuracy and robustness, particularly by reducing overfitting through ensemble methods. This study strongly influenced our decision to prioritize Random Forest in our experimentation, considering its balance between accuracy and generalization.
- **Paper 2: Comparative Analysis of Machine Learning Algorithms in Socioeconomic Data.** This study compares various machine learning models, including Neural Networks, Decision Trees, and SVMs, across several socioeconomic datasets, focusing on the trade-offs between accuracy and interpretability. Complex models like Neural Networks showed higher accuracy, while simpler models like Decision Trees were easier to interpret. The paper influenced our decision to test both complex models for accuracy and simpler models

for interpretability, ensuring that our model selection is well-rounded.

These findings underscore the importance of balancing model accuracy with interpretability. Based on the insights from these papers, we decided to focus on Random Forest and Logistic Regression as part of our experimental approach.

3. Dataset and Preprocessing

3.1. Dataset Overview

The dataset used in this study is the Census Income Dataset, containing 48,842 instances and 14 features related to demographic and employment data. The target variable is binary: income greater than or less than $50K per year.

Name	Type	Values	Description
age	Continuous	From 17 to 90	The age of an individual
workclass	Nominal	Private, Federal-Government, etc	A general term to represent the employment status of an individual
fnlwgt	Continuous	Integer greater than 0	Final weight: is the number of people the census believes the entry represents
education	Ordinal	Some-college, Prof-school, etc	The highest level of education achieved by an individual
education-num	Discrete	From 1 to 16	The highest level of education achieved in numerical form
marital-status	Nominal	Married, Divorced, etc.	Marital status of an individual
occupation	Nominal	Transport-Moving, Craft-Repair, etc	The general type of occupation of an individual
relationship	Nominal	Unmarried, not in the family, etc	Represents what this individual is relative to others
race	Nominal	White, Black, Hispanic, etc.	Descriptions of an individual's race
sex	Nominal	Male, Female	The biological sex of the individual
capital-gain	Continous	Integer greater than or equal to 0	Capital gains for an individual
capital-loss	Continous	Integer greater than or equal to 0	Capital loss for an individual
hours-per-week	Continous	From 1 to 99	The hours an individual has reported to work per week
native-country	Nominal	United-States, Cambodia, England, Puerto-Rico, Canada and more	Country of origin for an individual
income	Discrete	(≤50k USD, >50k USD)	The label whether or not an individual

Figure 1. Dataset Description

3.2. Data Preprocessing

1. **Data Cleaning**
   - Fixed income column by merging ("≤50K.", ">50K.") with ("≤50K", ">50K")
   - Replaced "?" with "Unknown" in 'workclass', 'occupation', and 'native-country'
   - Removed duplicate "education" column
   - Dropped 1200 rows with missing values
2. **Feature Engineering**
   - Applied label encoding to categorical features
   - Standard scaling for continuous numerical features

4. EDA

Exploratory Data Analysis (EDA) is vital for understanding data characteristics, identifying issues like class imbalance or outliers, and shaping model development strategies. The dataset comprises 48,842 records with 15 features, showing a class imbalance (76.07% earning ≤$50k vs. 23.93% earning >$50k). Balancing techniques may be needed to address this 3:1 ratio.

4.1. Histograms

Numerical features display varied distributions. **Age** is right-skewed (20-90 years), **education-num** has peaks at 9, 10, and 13 years, and **hours-per-week** peaks at 40, with secondary peaks at 20 and 60. Financial features (*fnlwgt*, *capital-gain*, *capital-loss*) are highly skewed with many zero values, suggesting potential transformations.

4.2. Pairplot

Key correlations include:

- **Age:** Positively correlates with *capital-gain* and *hours-per-week*.
- **Education-num:** Strongly correlates with income and *hours-per-week*.
- **Capital-gain:** Higher values cluster in the >$50K group.

Categorical features like *workclass*, *occupation*, and *marital-status* exhibit distinct clustering with income.

4.3. Box and Violin Plots

Strong predictors: *Education-num*, *capital-gain*, *age*, *hours-per-week*, and *marital-status/relationship* show clear differences across income groups.

Weak predictors: *Fnlwgt*, *race*, *native-country*, and *workclass* show minimal variations, making them less impactful in distinguishing income levels.

5. Methodology

5.1. Handling Class Imbalance

The dataset used in this study is imbalanced, with a higher proportion of instances labeled as earning more than $50k. To address this class imbalance, **Random Oversampling** was applied to the minority class, ensuring that the model had a more balanced representation of both classes. The oversampling technique duplicates examples from the minority class to equalize the number of instances between the two classes. Oversampling helps the model to learn from a more representative distribution, reducing the bias towards the majority class.

5.2. Train-Test Split

The dataset was split into a training set and a testing set, with 80% of the data used for training and 20% reserved for testing. This split helps to ensure that the model is evaluated on data it has not seen during training, providing an unbiased estimate of its performance on unseen data.

5.3. Feature Selection

Feature selection was implemented to enhance model efficiency and interpretability while maintaining predictive performance. A two-stage approach was used:

5.3.1 Feature Importance with Random Forest

Feature importance scores were extracted from the Random Forest model, highlighting:

- *fnlwgt* (15%) and *Age* (15%) as the most important features.
- *Relationship* (11%) and *Education-num* (10%) also showed significant influence.
- Features like *occupation*, *workclass*, and *capital-loss* had minimal impact (<4%).

Based on this, the top 10 features were selected for further analysis.

5.3.2 Recursive Feature Elimination (RFE)

RFE, with Random Forest as the base estimator, was applied to refine the selection. The final selected features included *age*, *workclass*, *fnlwgt*, *education-num*, *marital-status*, *occupation*, *relationship*, *capital-gain*, *capital-loss*, and *hours-per-week*.

5.3.3 Evaluation of Selected Features

The selected features were used to retrain Random Forest, MLP, and SVM models. No significant performance degradation was observed, confirming that dimensionality reduction improved computational efficiency without sacrificing accuracy or interpretability.

5.4. Model Selection

We implemented several models to assess their performance on predicting income levels, including:

- **Logistic Regression**: Chosen for its simplicity and interpretability, Logistic Regression provides a linear decision boundary and is useful for providing probabilities of outcomes.
- **Decision Trees**: Selected for their ability to model non-linear relationships and for easy interpretability, Decision Trees divide the data into homogenous subsets based on feature values.
- **Random Forest**: A powerful ensemble method that combines multiple decision trees to reduce overfitting by averaging predictions from individual trees. Random Forests are highly effective in handling both classification and regression problems.
- **Support Vector Machine (SVM)**: Chosen for its ability to handle non-linear decision boundaries and its strong theoretical foundation in maximum margin classification.
- **MLP Neural Networks**: Selected for their capacity to model complex relationships in the data through multiple layers of neurons, which allows for more flexibility in capturing intricate patterns.

5.5. Hyperparameter Tuning

Hyperparameter tuning was performed using **GridSearchCV** to optimize the performance of each model. The following optimal parameters were selected for the models:

- **Random Forest:** `bootstrap=False`, `max_depth=None`, `min_samples_leaf=1`, `min_samples_split=2`, achieving a best model score of 0.924.
- **Neural Network (MLP):** `activation='relu'`, `alpha=0.002`, `hidden_layer_sizes=(150, 150)`, `learning_rate='constant'`, `solver='adam'`, with a best model score of 0.827.
- **SVM:** `C=100`, `gamma=1`, `kernel='rbf'`, yielding a best model score of 0.884.

5.6. Model Evaluation

Model performance was assessed using multiple evaluation metrics, including:

- **Accuracy**: This represents the overall correctness of the model.
- **F1-Score**: A measure that combines precision and recall into a single metric, providing a balance between false positives and false negatives.
- **Precision**: Measures the proportion of positive identifications that were actually correct.
- **Recall**: Measures the proportion of actual positive cases that were correctly identified by the model.
- **Confusion Matrix**: A detailed matrix that visualizes the performance of the classification model by showing the true positive, false positive, true negative, and false negative results.
- **AUC-ROC Curve**: Measures the ability of the model to distinguish between classes, with the area under the curve indicating the model's discriminatory power.

5.7. K-Fold Cross-Validation

To further assess the performance of the models and ensure their generalizability, **K-Fold Cross-Validation** was performed. This method splits the dataset into K subsets and trains the model K times, each time using a different subset as the test set while the others are used for training. The average performance across all K iterations provides a more reliable estimate of model performance.

5.8. Boosting Algorithms with Random Forest

Boosting with Random Forest was also explored as a potential improvement to model performance. However, this approach led to a significant decrease in accuracy, resulting in poorer performance compared to the original Random Forest model. Despite the application of boosting techniques, such as AdaBoost and Gradient Boosting, the model failed to outperform the basic Random Forest classifier and was thus deprioritized.

5.9. Final Model Selection

After applying Random Oversampling and performing hyperparameter tuning, the **Random Forest** model emerged as the best-performing model. It outperformed other models like Logistic Regression, Decision Trees, MLP Neural Networks, and SVM in terms of accuracy, F1-Score, and robustness to overfitting. The Random Forest model's ability to handle imbalanced data and its consistent performance across cross-validation splits made it the final model selected for deployment.

6. Results and Analysis

6.1. Performance Metrics

The performance of the models after **Random Over Sampling**

- **Logistic Regression:**
  - *Accuracy:* 76.8% | *F1 Score:* 0.769

  Logistic Regression maintained consistent training and testing performance but underfitted the data.

- **Decision Tree:**
  - *Accuracy:* 91.6% | *F1 Score:* 0.921

  Despite oversampling, the Decision Tree showed significant overfitting, with near-perfect training performance but lower test performance.

- **Random Forest:**
  - *Accuracy:* 93.3% | *F1 Score:* 0.936

  The Random Forest model showed the best performance before hyperparameter tuning, with good accuracy and F1-score.

The performance of the models after **Hyperparameter Tuning**

- **Random Forest:**
  - *Accuracy:* 93.9% | *F1 Score:* 0.939

  After hyperparameter tuning, the Random Forest model showed slight improvement in accuracy and F1-score.

- **MLP Neural Networks:**
  - *Accuracy:* 82.8% | *F1 Score:* 0.827

  The MLP Neural Network performed well but did not surpass the Random Forest in terms of overall accuracy and F1-score.

- **SVM (RBF Kernel):**
  - *Accuracy:* 89.8% | *F1 Score:* 0.898

  SVM with an RBF kernel also showed good performance but still lagged behind Random Forest in both accuracy and F1-score.

6.2. K-Fold Cross-Validation Results

Random Forest model showed a mean cross-validation score of 0.92; SVM being a close competitor with mean cross-validation score as 0.88, followed by MLP at 0.82. The cross-validation results showed that the Random Forest model was consistently performing well across different subsets of the data, confirming its robustness and generalizability.

7. Conclusion

The primary outcome of this project is that the Random Forest model provides the best performance for predicting income levels. Among all the models tested, Random Forest achieved the highest accuracy, F1-score, and generalization, making it the most effective choice for this task. The model's robustness to overfitting, demonstrated through its consistent performance across training and test sets, further supports its suitability for this application.

Best Model: Random Forest

Despite the competitive performance of other models like SVM and MLP, Random Forest demonstrated superior results. The use of random oversampling and hyperparameter tuning optimized the model, ensuring that it is both well-calibrated and capable of generalizing effectively

Accuracy	F1 Score	Precision	Recall	Mean CV
0.9394	0.9393	0.9410	0.9394	0.9243

Table 1. Performance Metrics of Random Forest Model

across different data subsets. This makes it a highly reliable model for deployment in income prediction tasks, providing a strong foundation for future developments in this area. This shows that tree-based models, particularly Decision Trees and Random Forest, are highly effective for classification tasks involving complex socioeconomic data.

7.1. Individual Contributions

- **Sameer Singh Godara**: Visualization, Feature selection, Boosting Algorithm, Report writing, and Presentation preparation.
- **Sanyam Barwar**: Preprocessing, Exploratory data analysis (EDA), Final model training, Project management and finalization.
- **Sanyam Garg**: Data collection, Feature engineering, and Model comparison, Cross-Validation
- **Vivan Rangra**: Model Selection, Initial model training, Hyperparameter tuning

References

[1] Ronny Kohavi and Barry Becker (1996). *Adult Census Income Dataset*. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/20/census+income

[2] S. Shukla, A. Kumar and H. Kumar (2023). *Binary Classification of Adult Census Income Dataset: Analysis and Comparison of Machine Learning and Deep Learning Techniques*. IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 535-540. https://ieeexplore.ieee.org/document/10181907

[3] Alghazwi, M., Alqahtani, S., Almutlaq, N. and Alshathri, S. (2022). *Adult Income Prediction using Machine Learning Classification Techniques*. Procedia Computer Science, Vol. 207, pp. 2192-2201. https://www.sciencedirect.com/science/article/pii/S1877050922021159

[4] AlDhuwayhi, F. (2020). *Census Income Prediction*. GitHub Repository. https://github.com/Faisal-AlDhuwayhi/Census-Income-Prediction