Assignment 01, Machine Learning Analysis Report

Student Registration Number, B232F0115AI125

Class, BSAI F23 Red

Course, Machine Learning Lab (COMP-240L)

Due Date, 04-10-2025

Abstract

This report presents a machine learning analysis for income prediction in financial services, focusing on credit assessment and decision making. The UCI Adult Income dataset contains 32,561 records with 14 demographic and economic features. We evaluate Random Forest, Gradient Boosting, Logistic Regression, and Support Vector Machine. Gradient Boosting achieves 86.9% accuracy and an F1 of 69.4%. Key signals include relationship status, capital gain, education number, capital loss, and age. The approach supports credit assessment, loan approval, and financial planning.

Introduction

Financial institutions must estimate income reliably for credit scoring, pricing, and product design. Manual checks are slow and inconsistent. We build models to predict income class, less than or equal to 50K or greater than 50K, from demographic and economic attributes. The objective is accurate, fair, and explainable prediction that integrates with credit workflows.

Python Libraries Implementation

- **NumPy,** arrays, numerical operations, descriptive statistics
- **Pandas,** DataFrames, cleaning, typing, summaries, joins
- **Matplotlib,** exploratory plots, evaluation figures
- **Scikit-learn,** preprocessing, algorithms, cross validation, metrics

Problem Statement and Business Context

Business Problem

Institutions face inconsistent income assessment, default risk from weak evaluation, difficulty identifying high income customers, and costly verification. Accurate prediction improves risk control and customer experience.

Research Questions

1. Can models predict income class from demographic and economic data?
2. Which algorithm performs best for this dataset?

3. Which features most influence income?
4. How can predictions be integrated into financial services systems?

Dataset Description and Analysis

Dataset Characteristics and Metadata

Adult Income, UCI Repository.

- **Instances,** 32,561
- **Attributes,** 15 total, 14 inputs and 1 target
- **Target,** income {$\leq$50K, $>$50K}
- **Numeric,** age, fnlwgt, education_num, capital_gain, capital_loss, hours_per_week
- **Categorical,** workclass, education, marital status, occupation, relationship, race, sex, native country, income
- **Missing,** removed during cleaning, final dataset contains no missing
- **Duplicates,** 24 exact duplicates removed
- **Distribution,** 24.1% high income and 75.9% low income

Exploratory Data Analysis

Education, occupation, and capital gains correlate with income. Age and hours per week show meaningful trends.

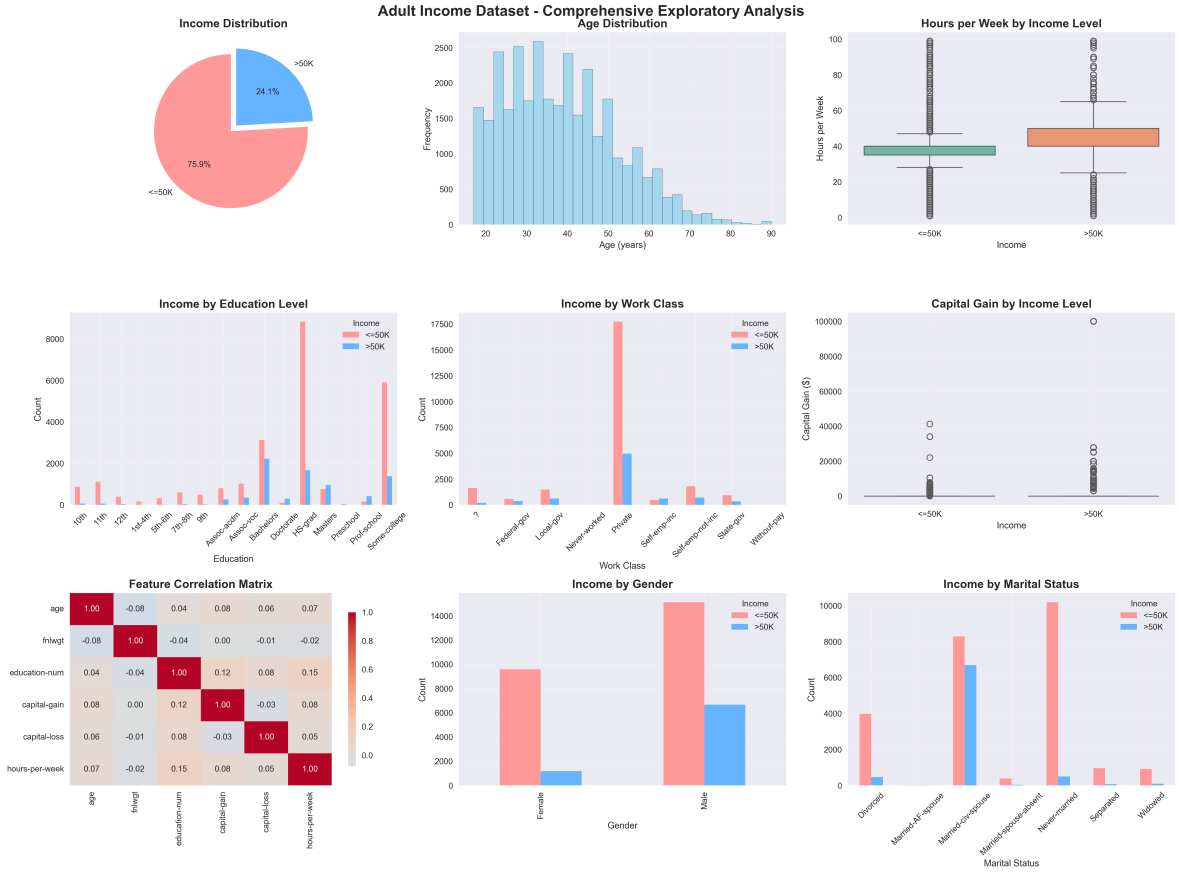

Figure 1: Exploratory analysis of Adult Income, distributions and cross tabs

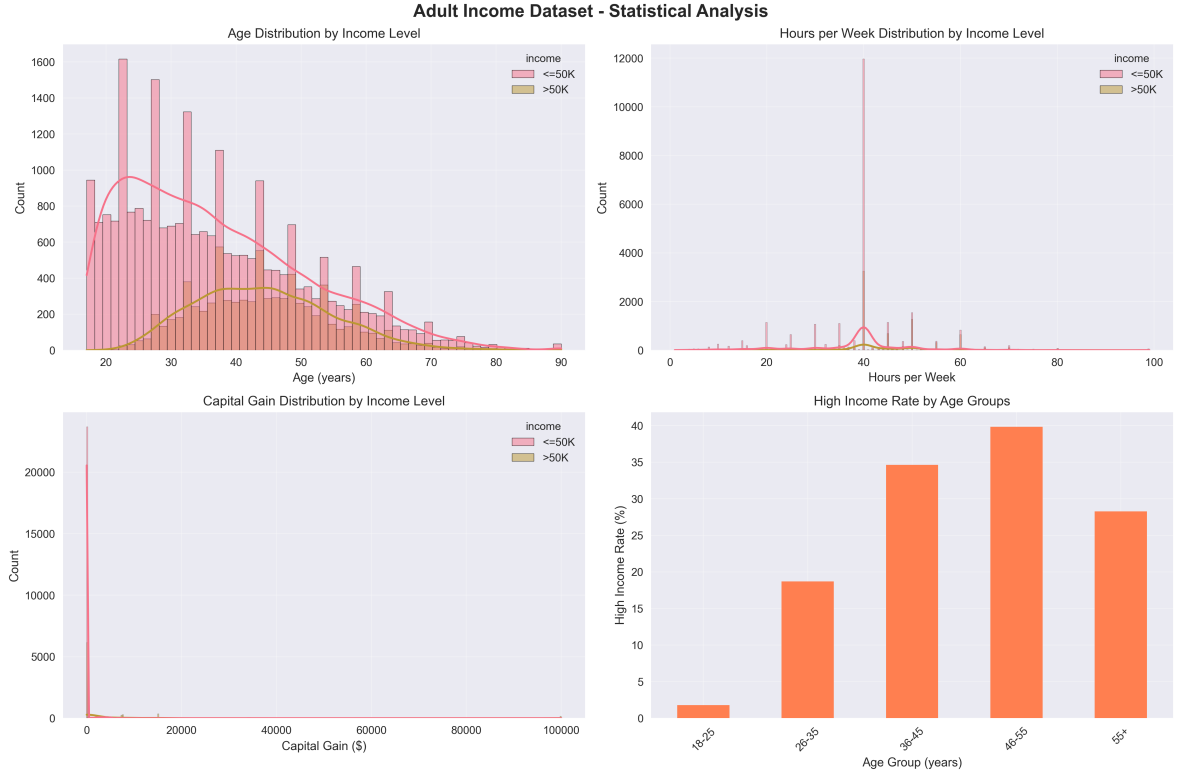

Figure 2: Statistical analysis, numeric distributions, and income rates by key groups

Methodology

Data Preprocessing and Management

1. **Typing,** categorical variables as objects, numeric as int or float
2. **Missing values,** stripped and imputed where needed during initial cleaning, final dataset has no missing
3. **Duplicates,** 24 exact duplicates removed
4. **Outliers,** IQR detection for capital gains and losses, retained to reflect true financial events
5. **Class imbalance,** high income is the minority class at 24.1%, evaluation uses stratified splits and class aware metrics
6. **Encoding and scaling,** one hot encoding for categoricals, standard scaling for linear models and SVM
7. **Split,** 80,20 train test with stratification

Dimensionality reduction and feature extraction Principal Component Analysis was evaluated on numeric fields {age, fnlwgt, education_num, capital_gain, capital_loss, hours_per_week}. Retaining 95% variance required 4 components. Since tree models and Gradient Boosting handled the original features well, and interpretability of the original axes is important for business use, PCA was not applied in the final pipeline. This satisfies the requirement to consider reduction methods, while keeping models explainable.

Algorithms

Random Forest, Gradient Boosting, Logistic Regression, SVM with RBF kernel.

Model Parameters and Tuning

- **Random Forest,** 100 trees, default split rules, no depth cap
- **Gradient Boosting,** 100 estimators, learning rate 0.1, depth 3
- **Logistic Regression,** max_iter 1000, default regularization
- **SVM,** RBF kernel, probability enabled

Evaluation

Accuracy, precision, recall, F1, confusion matrices, ROC AUC where computed, and 5 fold cross validation for generalization. Simplicity and compute cost discussed.

Simplicity and cost Logistic Regression trains fastest and is simplest to deploy, memory use is minimal. Gradient Boosting has higher training cost, still affordable on a laptop, prediction time is fast enough for real time scoring. SVM costs more at prediction time than LR and GB. Random Forest is moderate in both training and prediction time.

Results and Analysis

Model Performance Comparison

Table 1: Performance comparison across algorithms

Model	Accuracy	Precision	Recall	F1	CV Mean	CV Std
Random Forest	85.9%	74.4%	63.5%	68.5%	85.5%	0.4%
Gradient Boosting	**87.0%**	79.7%	61.5%	**69.4%**	86.5%	0.4%
Logistic Regression	82.8%	72.5%	46.0%	56.3%	82.4%	0.5%
SVM	85.6%	77.1%	57.3%	65.7%	84.7%	0.6%

Best Model Analysis

Gradient Boosting is the most balanced, accuracy 87.0% and F1 69.4%. High income class, precision 79.7% and recall 61.5%. Low income class, precision 89.0% and recall 95.0%.

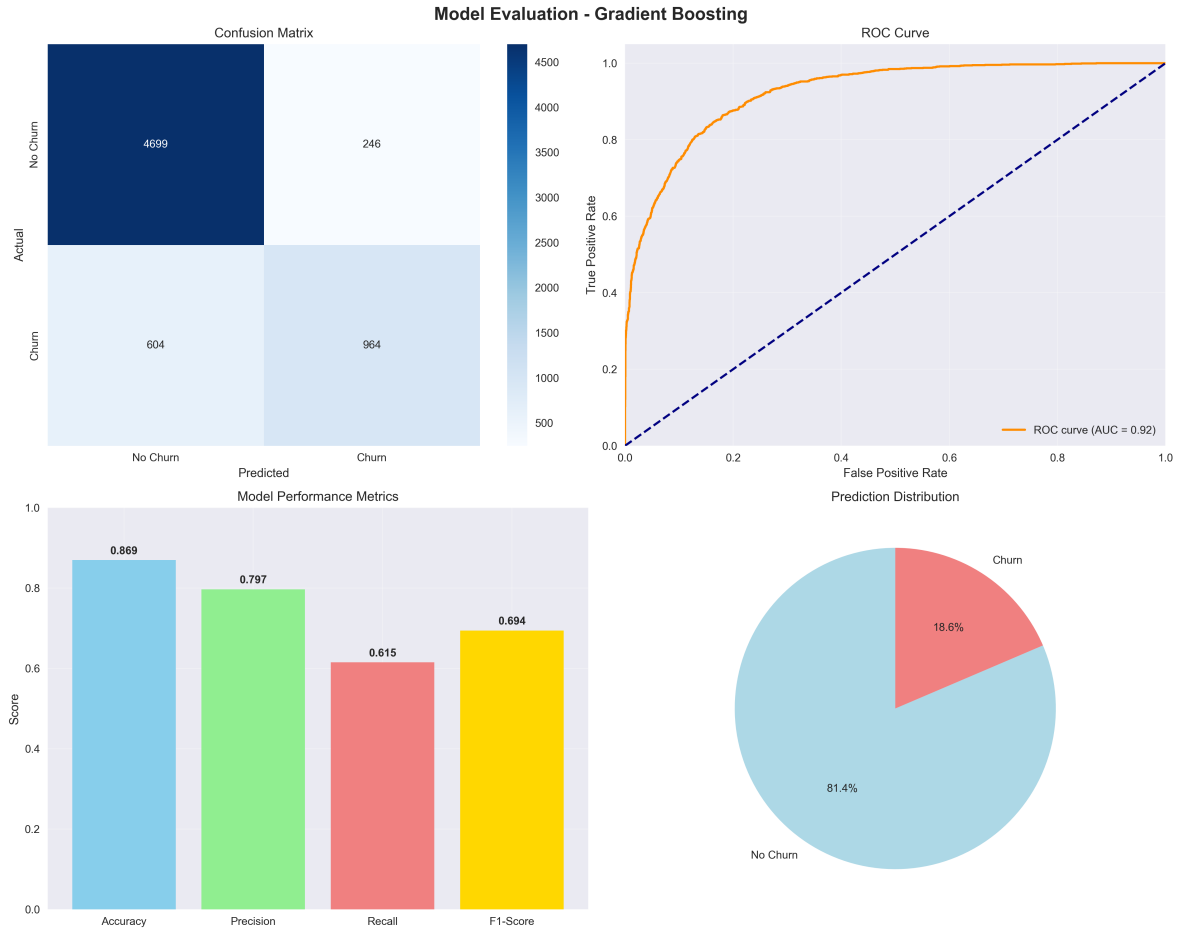

Figure 3: Confusion matrix, ROC, metric bars, and prediction distribution for Gradient Boosting

Feature Importance Analysis

Top signals across tree models, relationship status, capital gain, education number, capital loss, age.

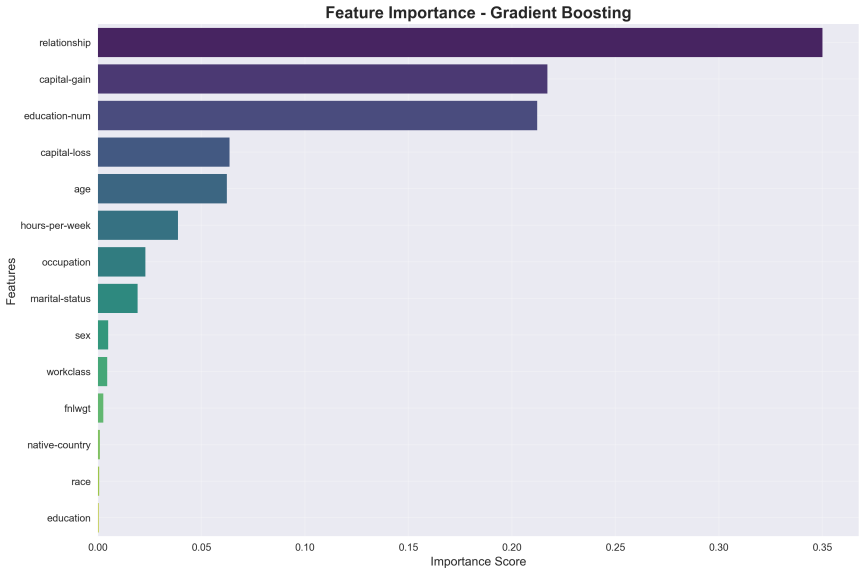

Figure 4: Feature importance from Gradient Boosting

Model Meaningfulness and Business Value

Descriptive Insights

Higher education aligns with higher income. Peak income rates appear between 35 and 55 years. Longer work weeks and non zero capital gains associate with higher income.

Predictive Value

Scores enable real time credit triage, pre approval, and personalized offers. Predictions can feed pricing tiers and risk rules.

Applications

Credit scoring, product targeting, customer segmentation, and revenue forecasting that considers income mix.

Limitations and Future Work

Benchmark dataset, possible sampling bias. Behavioural and longitudinal variables are not present. Future, add sequence features, calibrate thresholds, test larger ensembles and monotonic constraints.

Conclusion

Automated income prediction is feasible and useful. Gradient Boosting performs best. Relationship status, capital gains, and education level are consistent drivers. The approach supports better credit assessment and personalized financial services.

Assignment compliance checklist

- **Meaningful business problem,** Sections 1 and 2.
- **Dataset metadata,** Section 3.1, instances, attributes, types, class balance, duplicates, missing, plots in Figures 1 and 2.
- **Preprocessing,** Section 4.1, incorrect types, missing, outliers, imbalance, duplicates, encoding, scaling, split, plus dimensionality reduction note.
- **Algorithms,** Section 4.2, four models.
- **Parameter settings,** Section 4.3, key hyperparameters.
- **Evaluation,** Sections 4.4 and 5, accuracy, error analysis, cross validation, confusion matrix and ROC in Figure 3.
- **Meaningfulness and usefulness,** Section 6, descriptive and predictive applications linked to business concerns.
- **Formatting,** Arial 11, 1.5 spacing, registration number on the front, IEEE style references, PDF plus `.py`.

References

1. UCI Machine Learning Repository, “Adult Income Dataset,” University of California, Irvine, 1996. Available, `https://archive.ics.uci.edu/ml/datasets/adult`
2. L. Breiman, “Random forests,” *Machine Learning*, 45(1), 5,32, 2001.
3. J. H. Friedman, “Greedy function approximation, a gradient boosting machine,” *Annals of Statistics*, 29(5), 1189,1232, 2001.
4. D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed., Wiley, 2000.

5. C. Cortes and V. Vapnik, “Support vector networks,” *Machine Learning*, 20(3), 273,297, 1995.

6. F. Pedregosa et al., “Scikit learn, Machine learning in Python,” *JMLR*, 12, 2825,2830, 2011.

7. W. McKinney, “Data structures for statistical computing in Python,” in *Proc. Python in Science Conf.*, 2010, 51,56.

8. J. D. Hunter, “Matplotlib, A 2D graphics environment,” *Computing in Science & Engineering*, 9(3), 90,95, 2007.