Predicting Income Levels Using Machine Learning: A Classification Approach with the Adult Census Dataset

Hugo Ribeiro 113402 - hugo.ribeiro04@ua.pt
Pedro Ponte 98059 - ponte@ua.pt

***Abstract*—Income prediction has significant social and economic implications. This study explores the application of machine learning techniques to classify income levels using the UCI Adult Census dataset. We investigate three real-world use case scenarios high-risk loan approval, IRS auditing, and social security policies. Our methodology includes extensive preprocessing, feature engineering, and evaluation of classification models such as Logistic Regression, Random Forest, and XGBoost under various sampling strategies. The best results were achieved with XGBoost, which demonstrated strong performance across all scenarios. Furthermore, we incorporate fairness-aware adjustments aligned with ethical AI guidelines.**

***Index Terms*—Machine Learning, Income Classification, Census Data, Fairness in AI**

I. Introduction

INCOME prediction is an important area of study with a lot of implications for economic policy, social equity, and financial planning. The paper by Matkowski et al. [1] demonstrates that machine learning approaches can significantly outperform traditional prediction methods when predicting individual income levels. Their research, with data from the *Current Population Survey* dataset with 467,811 observations across 264 variables, provides strong evidence for the integration of machine learning models alongside the regular economic analysis techniques.

The ability to accurately predict income has great implications for addressing income inequality, which remains one of the most deep socioeconomic challenges in modern society. By developing more precise prediction models, researchers and policymakers can better understand the secondary factors that cause economic differences and plan more effective interventions.

Machine learning models offer advantages in this domain due to their capacity to identify complex patterns and relationships in large scale demographic and employment datasets. These models can process a lot of variables simultaneously and capture non-linear interactions that traditional statistics might miss, resulting in more accurate predictions of individual earning potential [2].

By addressing this prediction challenge, this project aims to contribute with methodological insights in the application of machine learning in economic forecasting and a deeper understanding of the determinants of income levels in modern society. The findings have potential applications across multiple domains, including policy development, workforce planning, financial services, and social research.

Rather than focusing on achieving the best overall performing models, we'll instead take a closer look into three use cases:

1) High-risk loan approval
2) IRS auditing
3) Eligibility for social security benefits

and explain our approach for optimizing to specific goals.

II. State of the Art

The use of machine learning for income prediction has evolved significantly in recent years, with various approaches showing great improvements over traditional economic modeling techniques. This section demonstrates some examples of this, with particular focus on approaches relevant to classification tasks using census data.

A. Machine Learning Approaches vs. Traditional Methods

Recent research consistently shows that machine learning algorithms outperform conventional econometric approaches in income prediction tasks. Matkowski et al. [1] demonstrated that machine learning models achieved superior performance compared to traditional prediction approaches when applied to data from the Current Population Survey containing 467,811 observations across 264 variables. Their research supports the integration of machine learning methodologies alongside traditional techniques in economic research focused on prediction.

Decision trees have also sown to be effective for income classification tasks. Asaduzzaman et al. [3] found that decision tree models outperformed other approaches when predicting whether employee salaries exceeded $50,000 using census data with 32,561 records. Similarly, Wang [4] reported that Random Forest (RF) algorithms achieved an accuracy of 98% in income classification tasks, with K-Nearest Neighbors (KNN) also performing strongly with 90% accuracy.

B. Feature Selection and Engineering

Feature selection plays a crucial role in improving model efficiency and predictive power. Lazar [5] employed Principal Component Analysis (PCA) alongside Support Vector Machines (SVM) to generate and evaluate income prediction data based on the Current Population Survey. This approach achieved accuracy values as high as 84% while reducing computational time by 60% through targeted feature selection.

Demographic variables have been identified as particularly important predictive factors. Wang [4] determined that age showed the highest relevance to potential income with an

importance score of 0.225 using Gini Importance (or Mean Decrease in Impurity) measures. Other significant features reported include education level, occupation, marital status, and hours worked per week, which aligns with what was expected about income determinants.

C. Advanced Techniques and Ensemble Methods

Ensemble methods have demonstrated considerable success in income prediction tasks. Random Forests and Gradient Boosted Trees (GBT) consistently outperform single-model approaches in classification accuracy [4]. Ellum et al. [6] got F1 scores up to 98.94% using Multi-Layer Perceptron Classifiers in related census data applications, showing good robustness.

More advanced approaches keep appearing, for example Echevin et al. [7] used deep neural networks with semi-supervised learning techniques like pseudo-labeling to improve prediction accuracy, achieving AUC scores between 0.8 and 0.9. This represents an improvement in using partially labeled datasets, which are often a common limitation in economic research.

D. Addressing Bias and Fairness

As interest in machine learning applications in this topic is increasing, researchers need to pay attention to mitigating potential biases. Zhang et al. [8] presented a framework for addressing unwanted biases through adversarial learning when working with census data. Their approach maintained accuracy while getting close to equality of odds, shwing that fairness considerations can be considered without significant performance penalties.

Chockalingam et al. [9] emphasized the importance of the analysis of exploratory data and feature understanding to develop models that accurately capture patterns without reinforcing problematic biases present in historical data. Their work achieved 87% accuracy while focusing on identifying the most important features explaining variance in income levels.

E. Current Limitations and Challenges

Despite significant progress in the technology area, there are still challenges to overcome, as many studies rely on relatively small or outdated datasets, potentially limiting their adaptation to current economic conditions. Additionally, the *black-box* nature of some high performing algorithms like deep neural networks and ensemble methods can limit interpretability, which is often crucial for economic policy applications, where bias can play an important part.

Furthermore, most models tend to classify income into two groups (above or below a certain amount) instead of predicting more detailed income ranges or exact values. The challenge is to make these models easier to understand while still keeping their predictions accurate.

III. Data description, visualization and statistical analysis

A. Problem Description

The problem addressed in this analysis is to predict whether an individual's income exceeds \$50k per year based on census data. This binary classification task uses the Adult Income dataset, which contains demographic and employment information collected by the US Census Bureau.

B. Dataset Overview

Our dataset contains 14 features about individuals, including demographic information (age, race, sex), education level, employment details (occupation, hours worked), and other socioeconomic indicators.

Features included:

- **Demographic attributes** - age, race, sex, native-country
- **Education indicators** - education level (categorical) and education-num (numerical equivalent)
- **Employment data** - workclass, occupation, hours-per-week
- **Financial indicators** - capital-gain, capital-loss
- **Others** - marital-status, relationship, fnlwgt (final weight)

The target variable **income** is binary: $>$50k for income above \$50k per year and $\leq$50k for income below or equal to \$50k per year.

C. Data Visualization and Statistical Analysis

1) Missing Values: The initial data analysis revealed missing values in three features:

- workclass - 1836
- occupation - 1843
- native-country - 583

These missing values represent approximately 7.41% of the dataset. Rather than dropping these rows, the missing values were imputed using the most frequent value in each column, which preserves more data while introducing minimal bias.

2) Distribution of Key Features: Some of our key features were distributed in groups for easier classification.

a) Age Distribution: Age ranges are divided in 10 year intervals, starting with [17, 26], [27,36], ... [87-99], with the most represented group being [27, 36] with a *fnlwgt* of 0.271

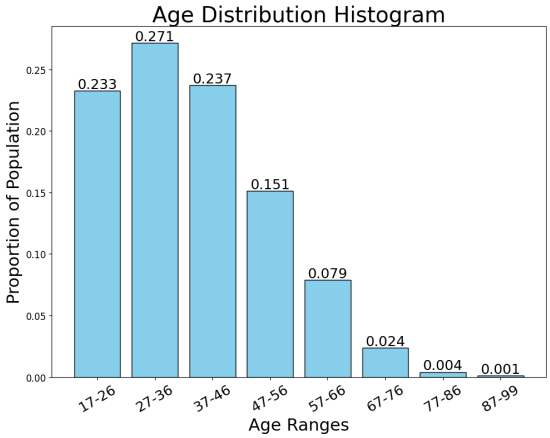

Fig. 1: Age Distribution Histogram

b) Education Distribution: High school graduates constitute the largest group (32.1%), followed by those either with Bachelor's degrees (16.3%) or Some-college (22.3%).

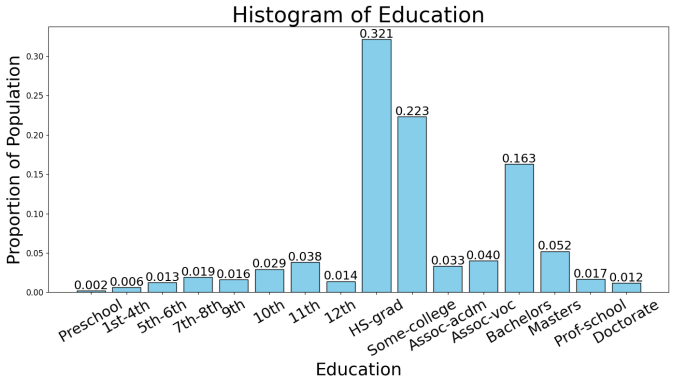

Fig. 2: Education Distribution Histogram

c) Race Distribution: The dataset is predominantly White (84.3%), with Black individuals comprising 11.7%.

d) Sex Distribution: Males are more represented (67.6%) compared to females (32.4%).

e) Income Distribution: As expected we can see a class imbalance with approximately 76% of individuals earning $\leq$\$50K.

f) Hours per Week Distribution: Most individuals (47.5%) work exactly 40 hours per week, with 28.7% falling in the *overtime* category for working more than 40 hours.

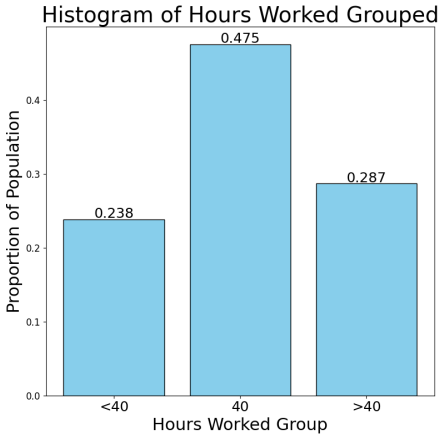

Fig. 3: Work Hours Histogram

D. Feature Analysis and Transformation

1) Capital Gain and Capital Loss: These features were highly skewed:

- Most values were zero (no capital gains/losses)
- Non-zero values had widely varying magnitudes

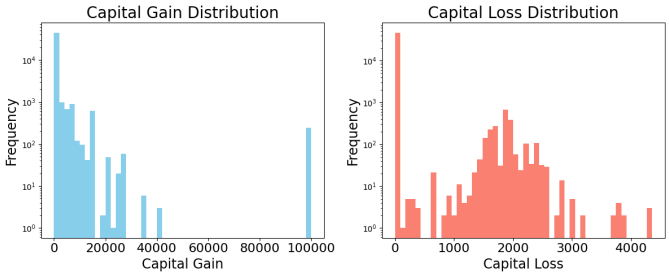

Fig. 4: Capital Gain and Capital Loss Distributions

To address this, we binned these continuous values into categories:

- none (zero)
- low (small amounts)
- high (large amounts)

This transformation helped **reduce noise** from extreme values while **preserving meaningful patterns**.

2) Hours per Week: This feature was transformed into three categories:

- part-time ($\leq$30 hours)
- full-time (31-40 hours)
- overtime ($>$40 hours)

This binning created more interpretable categories

E. Data Quality Observations

1) **Class Imbalance**: The target variable shows a significant imbalance with 76% in the $\leq$50k class and 24% in the $>$50k class.
2) **Demographic Imbalance**: The dataset shows notable imbalances in race (85.7% White) and sex (67% Male).
3) **Missing Values**: Three features had missing values that required imputation.
4) **Skewed Distributions**: Capital gain and capital loss features were highly skewed with many zero values.
5) **Feature Correlation**: Some features like education and education-num were directly related and potentially redundant.

IV. Dataset Preprocessing

This section provides an overview of the key methodological components involved in building our machine learning models for income prediction applied to our use cases. These components include data preprocessing, model selection and implementation, use case scenario definition, and hyperparameter tuning. Proper handling of these stages is crucial to ensure model performance, generalizability, and interpretability, especially when working with complex socio-economic datasets like the Adult Dataset, which contains diverse demographic and employment-related features.

Preprocessing is the foundation of any successful machine learning workflow. Real-world datasets like the Adult Dataset often contain missing values, inconsistent formats, and categorical variables that require careful handling.

A. Missing Values

As previously discussed, the dataset contains missing values in the workclass, occupation, and native-country features. We initially considered two common approaches to handle these missing entries: (1) removing all rows containing missing values, or (2) imputing the missing values using the mode (most frequent value) for each respective feature. We decided for the second approach upon the fact that approximately 7.41% of the total observations had missing values. Given that our dataset is imbalanced, with the income class (>50K) being underrepresented, removing rows would risk further reducing the representation of this class.

Feature	Most Common Value	% Frequency
workclass	Private	73.64%
occupation	Prof-specialty	13.41%
native-country	United-States	91.35%

TABLE I: Most Common Values Used for Mode Imputation

To support our choice, it is observed that two of the three features, workclass and native-country, have a relatively high frequency percentage for their most common values (73.64% and 91.35%, respectively). This indicates that the mode is a strong representative value for these features, making imputation a reasonable and minimally disruptive strategy.

B. Encoding Categorical Features

Machine learning models typically require numerical input, making it necessary to convert categorical variables into data our models could use. For this purpose, we applied one-hot encoding to features without an inherent ordinal relationship (e.g., marital-status, occupation), ensuring that the model does not assume any unintended order. For features with a natural hierarchy or ranking, we would have used label encoding to preserve their ordinal structure. The reason we did not use label encoding lies on the fact that the feature that would require it, **education**, was already encoded in the dataset as **education_num**.

One-Hot Encoded Features
workclass
marital-status
occupation
relationship
race
sex
native-country

TABLE II: One-Hot Encoded Features

Encoding categorical variables transforms each category into its own binary column, allowing machine learning models to interpret them numerically. For example, the original race feature is expanded into multiple binary columns such as race_AmerIndianEskimo, race_AsianPacIslander, race_Black, race_Other, and race_White. After applying one-hot encoding to the selected features, the dataset's dimensionality increased, resulting in a final shape of 90 columns.

We chose to retain all dummies, instead of omitting the first dummy for each encoding, which is a common approach. This decision was made because including all encoded categories improved the performance of logistic regression, while having no negative impact on tree-based models like XGBoost and Random Forest.

Lastly, the target variable was binarized to 0 ($\leq$50k) and 1 (>50k), facilitating binary classification.

C. Transforming Numerical Features

Some continuous numerical features, capital-gain, capital-loss and hours-per-week, exhibit extreme or highly granular values, which can introduce noise and reduce model generalization. To address this, we applied a binning strategy to simplify certain variables and enhance interpretability.

- **capital-gain** and **capital-loss** contain many zero entries and rare, high outliers. We grouped these into three categories:
  - **none**: for zero values,
  - **low**: for values up to a defined threshold,
  - **high**: for values above that threshold.

  Thresholds were selected based on data distribution: **7,000** for **capital-gain** and **2,000** for **capital-loss**.
- **hours-per-week** ranges from 1 to 99 and varies widely across individuals. We categorized it into work intensity groups:
  - **part-time**: $\leq$ 30 hours,
  - **full-time**: 31–40 hours,
  - **overtime**: $>$ 40 hours.

This binning approach helps reduce the effect of outliers, improves model stability, and provides more interpretable groupings of labor patterns and income components.

The resulting binned categories were subsequently one-hot encoded to convert them into a numerical format suitable for model input, ensuring consistency with the rest of the dataset's preprocessing pipeline.

D. Two Versions of the Dataset

Since binning alters the feature representation, we treated it as an optional preprocessing step. To assess its impact, we created **two versions of the dataset**:

1) One retaining the **original continuous numerical features**. (df1)
2) Another using the **binned and one-hot encoded versions** of those features. (df2)

This parallel dataset structure allows us to *compare model performance* under both scenarios and determine whether binning improves accuracy or interpretability.

E. Splitting and Sampling Strategy

To enable robust model evaluation and better simulate real-world application scenarios, we chose to ignore the predefined train/test split that came with the dataset. Instead, we generated multiple random 80/20 train-test splits for both versions of the dataset. This approach not only provides a more flexible

evaluation framework but also ensures consistency in performance measurement across different data partitions. We adopted a **stratified splitting** strategy to preserve the original class distribution in both the training and testing sets, ensuring fair and representative performance comparisons.

In addition to standard splits, we addressed the **class imbalance** present in the income target variable where the majority of records represent individuals earning $\leq$\$50K by generating multiple variations of the training data. Specifically, we considered the following 3 strategies:

- **Raw splits (R):** No resampling applied.
- **Oversampling (O):** Replication of minority class samples using **RandomOverSampler**.
- **Undersampling (U):** Reduction of majority class samples using **RandomUnderSampler**.

We implemented 10 randomized splits per dataset version, using different random_state seeds for diversity. This methodology allows us to compare model behavior and stability under different resampling techniques and ensures fair evaluation across models trained on both the original and binned datasets.

Example Split Sizes:

Split Type	**df1** (Original)	**df2** (Binned)	**Features**
Raw Split (Train)	39,073	39,073	89 / 95
Oversampled (Train)	59,448	59,448	89 / 95
Undersampled (Train)	18,698	18,698	89 / 95

TABLE III: Sizes of training sets in the first split for both datasets

The original dataset (df1) contains fewer features and observations compared to the binned version (df2), which includes additional encoded columns resulting from binning and transformation. Oversampling significantly increases the training set size by replicating minority class instances, while undersampling reduces it by discarding majority class examples. This variation in dataset sizes across strategies helps evaluate the trade-off between data quantity and class balance.

V. Model Training and Testing

A. Model Implementation

In this section, we implement and evaluate various machine learning models to predict whether an individual's income exceeds \$50K based on the dataset features. The primary objective is to apply classification techniques discussed in class, deepen our practical understanding, and identify the most effective model for this binary classification problem.

The models are trained on both versions of the dataset (original and binned) using different sampling strategies to assess their behavior under varying data distributions.

B. Evaluation Metrics

To ensure a comprehensive assessment of model performance, we use the following classification metrics:

- **Accuracy** — Proportion of total correct predictions. Effective on balanced datasets.
- **Precision** — Proportion of predicted positives that are actually positive. Important when false positives are costly.
- **Recall** — Proportion of actual positives that are correctly predicted. Critical when minimizing false negatives.
- **F1-Score** — Harmonic mean of precision and recall. Useful when a balance between the two is desired.
- **ROC AUC Score** — Measures the model's ability to distinguish between classes at different thresholds. Higher values indicate better general performance.

C. Implemented Models

We selected three classification algorithms with varying complexity and characteristics to compare performance across different data representations and sampling techniques:

- **Logistic Regression** — A simple and interpretable linear model that serves as a strong baseline for binary classification.
- **Random Forest Classifier** — A powerful ensemble method using decision trees.
- **Gradient Boosting (XGBoost or LightGBM)** — A high-performance boosting algorithm that captures complex patterns and often yields state-of-the-art results on structured datasets.

Dataset	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
df1_O	0.81 ± 0.00	0.56 ± 0.01	0.84 ± 0.01	0.67 ± 0.01	0.90 ± 0.00
df1_U	0.80 ± 0.00	0.56 ± 0.01	0.84 ± 0.01	0.67 ± 0.01	0.90 ± 0.00
df1_R	0.85 ± 0.00	0.73 ± 0.01	0.60 ± 0.01	0.66 ± 0.01	0.90 ± 0.00
df2_O	0.81 ± 0.00	0.57 ± 0.01	0.84 ± 0.01	0.68 ± 0.01	0.91 ± 0.00
df2_U	0.81 ± 0.00	0.57 ± 0.01	0.85 ± 0.01	0.68 ± 0.01	0.91 ± 0.00
df2_R	0.85 ± 0.00	0.75 ± 0.01	0.59 ± 0.01	0.66 ± 0.01	0.91 ± 0.00

TABLE IV: Logistic Regression Performance (mean ± std)

Dataset	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
df1_O	0.85 ± 0.00	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	0.90 ± 0.00
df1_U	0.81 ± 0.00	0.58 ± 0.01	0.83 ± 0.01	0.68 ± 0.01	0.90 ± 0.00
df1_R	0.85 ± 0.00	0.73 ± 0.01	0.62 ± 0.01	0.67 ± 0.01	0.90 ± 0.00
df2_O	0.84 ± 0.00	0.65 ± 0.00	0.66 ± 0.01	0.66 ± 0.00	0.89 ± 0.00
df2_U	0.81 ± 0.00	0.57 ± 0.01	0.81 ± 0.01	0.67 ± 0.00	0.90 ± 0.00
df2_R	0.84 ± 0.00	0.70 ± 0.00	0.60 ± 0.01	0.65 ± 0.01	0.89 ± 0.00

TABLE V: Random Forest Performance (mean ± std)

Dataset	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
df1_O	0.84 ± 0.00	0.61 ± 0.01	0.85 ± 0.01	0.71 ± 0.00	0.93 ± 0.00
df1_U	0.83 ± 0.00	0.60 ± 0.01	0.87 ± 0.01	0.71 ± 0.00	0.93 ± 0.00
df1_R	0.87 ± 0.00	0.78 ± 0.01	0.65 ± 0.01	0.71 ± 0.01	0.93 ± 0.00
df2_O	0.82 ± 0.00	0.59 ± 0.01	0.83 ± 0.01	0.69 ± 0.00	0.91 ± 0.00
df2_U	0.81 ± 0.00	0.57 ± 0.01	0.85 ± 0.01	0.68 ± 0.00	0.91 ± 0.00
df2_R	0.86 ± 0.00	0.74 ± 0.01	0.62 ± 0.01	0.68 ± 0.01	0.91 ± 0.00

TABLE VI: XGBoost Performance (mean ± std)

VI. Model Performance Summary

The performance differences between the df1 and df2 dataset variants are generally within 1–2% across all models and metrics. These changes are minor and do not significantly affect overall model behavior or conclusions, meaning the variations between the two dataset versions can be considered negligible in practice. This occurs because logistic regression

tends to generalize well and is less sensitive to feature sparsity or small structural changes in the data. In contrast, tree-based models like Random Forest and XGBoost often benefit from more distinct or sparse features to create sharper splits, but even for them, the losses here are minimal. Given the added complexity and preprocessing time, we recommend skipping this step in future implementations.

A. Logistic Regression

Logistic Regression consistently provides **high recall** (0.84–0.85) across oversampled and undersampled data, making it a solid option when minimizing false negatives is essential. However, its **precision is consistently low** (0.56–0.57), which limits its F1 Score. On raw data, precision improves (0.73), but recall drops, indicating that oversampling is beneficial for this model when high recall is a priority.

B. Random Forest

Random Forest is competitive across all metrics and performs especially well in **accuracy** and **precision** on raw datasets (**df1_R: Accuracy = 0.85**, **Precision = 0.73**). While it benefits somewhat from oversampling, it slightly underperforms XGBoost in terms of ROC AUC (0.90 vs. 0.93). Undersampling shifts performance toward higher recall but compromises overall precision and accuracy due to reduced training data.

C. XGBoost

XGBoost achieves the highest overall performance, particularly on raw datasets (**df1_R**, **df2_R**), with ROC AUC scores around **0.93**. Precision and F1 Score also peak on raw data (e.g., **Precision: 0.78**, **F1: 0.71** on **df1_R**), indicating that XGBoost performs best when the original class distribution and feature richness are preserved. Oversampling improves recall slightly but tends to reduce precision, while undersampling provides negligible benefit.

VII. Impact of Sampling Techniques

Sampling	Pros	Cons
Oversampling (O)	Improves recall across models; maintains moderate F1 performance	Slightly reduces precision due to duplication and overfitting risk
Undersampling (U)	Boosts recall in Logistic Regression and XGBoost	Reduces overall precision and accuracy due to data loss
Raw (R)	Best precision and ROC AUC, especially for tree-based models	Lower recall; may miss minority class examples

TABLE VII: Comparison of Sampling Strategies

VIII. Use Cases & Hyperparameter Tuning

We'll now take a closer look at our actual models and how they were applied to help in real-world problems.

A. Hyperparameter Tuning Approach

To optimize model performance, we employed a comprehensive hyperparameter tuning strategy using `GridSearchCV`. The tuning process focused specifically on maximizing **relative performance for each use case**. We defined a parameter grid encompassing key XGBoost hyperparameters, including the number of estimators, tree depth, learning rate, and subsampling ratios for both rows and columns. A `StratifiedKFold` cross-validation strategy with five folds, shuffling, and a fixed random seed ensured robust and balanced evaluation across splits. The best hyperparameter combination was selected based on the highest average precision score obtained from cross-validation on the training data. These hyperparameters were later applied on the models and performance was assessed.

B. *High-risk loan approval*

The first case is for high-risk loan approval, this was considered when thinking about financial institutions which have to decide between approving or refusing credit applications. This task is considered a risk in most cases as institutions only have current and past information about the applicant and it can be hard to evaluate how a certain individual's future status, since some careers are more promising than others.

Here our focus was on maximizing **precision** so we had a very high confidence that the loan would be paid back, minimizing the risk of losing money.

a) Model chosen: The best performing model for this task was **XGBoost** which used the dataset ***df1_raw***.

b) Results: For this model we achieved the following values which show our predicted high precision, with a *false positive* count of only 12 (0.16%). This comes with the cost of missing a lot of entries that were actually >50k, with the *false negatives* being 1786. Our metrics before and after the tuning were the following:

Setting	Accuracy	Precision	Recall	F1 Score	ROC AUC
Before	0.87 ± 0.00	0.78 ± 0.01	0.65 ± 0.01	0.71 ± 0.01	0.93 ± 0.00
After	0.82 ± 0.00	0.98 ± 0.00	0.24 ± 0.01	0.38 ± 0.01	0.89 ± 0.00

TABLE VIII: XGBoost **df1_raw** Performance Before and After Precision-Oriented Tuning (mean ± std)

Although all the metrics were negatively impacted by our tuning, the precision saw a substantial increase which serves our purpose for this use case.

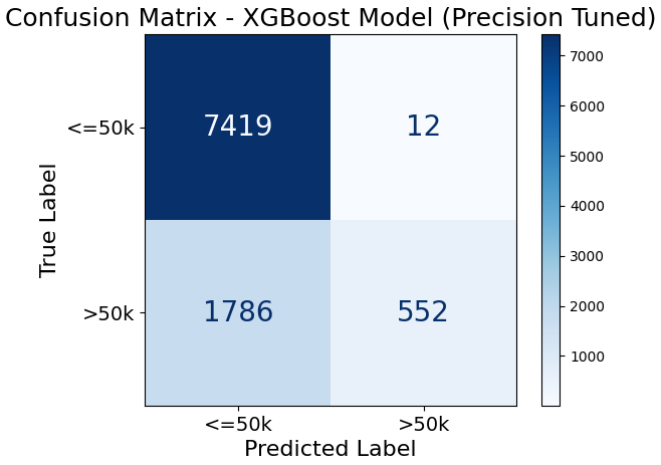

Fig. 5: Loan Approval Confusion Matrix

True Negatives	False Positives	False Negatives	True Positives
7419	12	1786	552

TABLE IX: XGBoost **df1_raw** Precision Tuned Confusion Matrix

C. *IRS auditing*

Another use of our models is tracking which individuals should be looked into by the IRS for trying to forge their income. By focusing on **recall** for the **>50k class** we can create a group of individuals with higher incomes and which should be more strictly audited for possible tax fraud.

a) Model chosen: For this use case the best performer was **XGBoost** with ***df1_UnderSampling***

Setting	Accuracy	Precision	Recall	F1 Score	ROC AUC
Before	0.83 ± 0.00	0.60 ± 0.01	0.87 ± 0.01	0.71 ± 0.00	0.93 ± 0.00
After	0.78 ± 0.01	0.52 ± 0.01	0.88 ± 0.01	0.66 ± 0.01	0.90 ± 0.00

TABLE X: XGBoost **df1_UnderSampling** Performance Before and After Recall-Oriented Tuning (mean ± std)

In this case the tuning was not optimal and even though we improved our recall by 1%, all other metrics got worse.

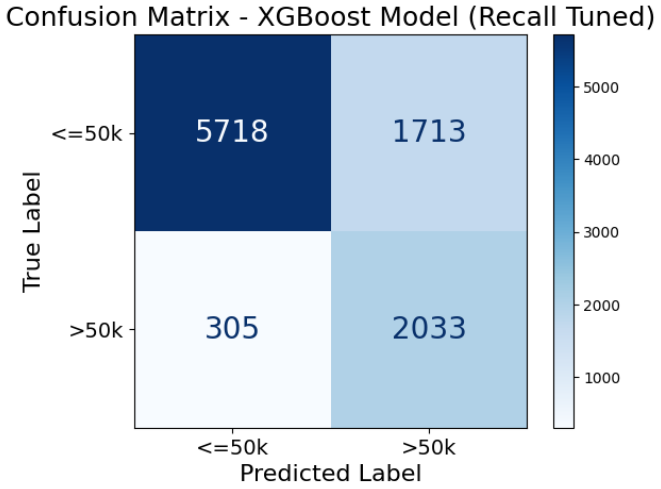

Fig. 6: IRS Auditing Confusion Matrix

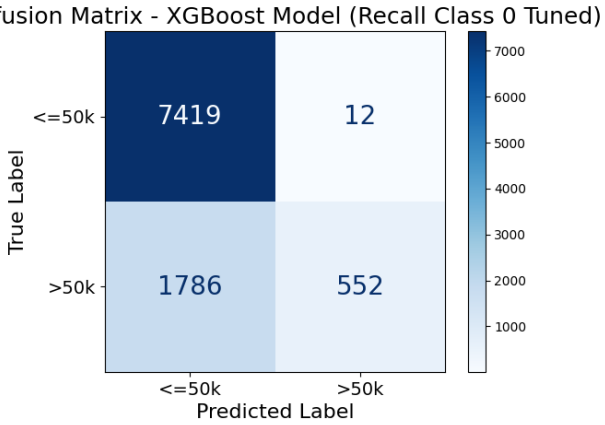

Fig. 7: Benefits Elegibility Confusion Matrix

True Negatives	False Positives	False Negatives	True Positives
5718	1713	305	2033

TABLE XI: XGBoost **df1_UnderSampling** Recall Tuned Confusion Matrix

D. *Eligibility for social security benefits*

Another important example on how these models can help society is better identifying individuals who may need economic help. This can be done by optimizing our model for a high recall for the ≤50k class, ensuring no one is left behind, even if that comes with a higher cost.

It should be noted that this and the Loan Use case use the same model, but with the target labels inverted. This resulted in many similarities like accuracy, ROC-AUC feature importance.

Setting	Accuracy	Precision	Recall	F1 Score	ROC AUC
Before	0.87 ± 0.00	0.90 ± 0.00	0.94 ± 0.00	0.92 ± 0.00	0.93 ± 0.00
After	0.82 ± 0.00	0.81 ± 0.00	1.00 ± 0.00	0.89 ± 0.00	0.89 ± 0.00

TABLE XII: XGBoost **df1_raw** Class 0 Performance Before and After Tuning (mean ± std)

For the parameters found we managed to improve an already good recall to a nearly perfect one, at little cost on other metrics.

The values registered here are the same the one found in **Use case 1** since the parameters are the exact same. Regardless, we can see that almost no one would be left behind, making this model great for helping those in need.

True Negatives	False Positives	False Negatives	True Positives
7419	12	1786	552

TABLE XIII: XGBoost **df1_raw** Recall Class 0 Tuned Confusion Matrix

IX. Fairness Analysis

To ensure responsible and unbiased decision-making, we conducted a fairness analysis by examining the most influential features across our three different use cases.

The top five most influential features for three different use cases, Loan Case, IRS Investigation Case, and Social Security

Case, along with their corresponding importance scores are summarized below:

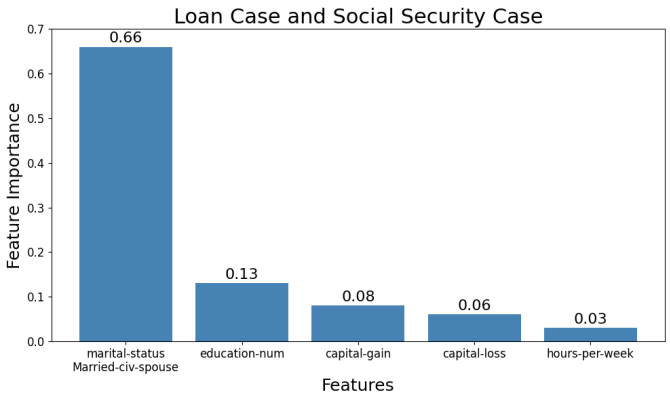

Fig. 8: Loan and Social Security Fairness Histogram

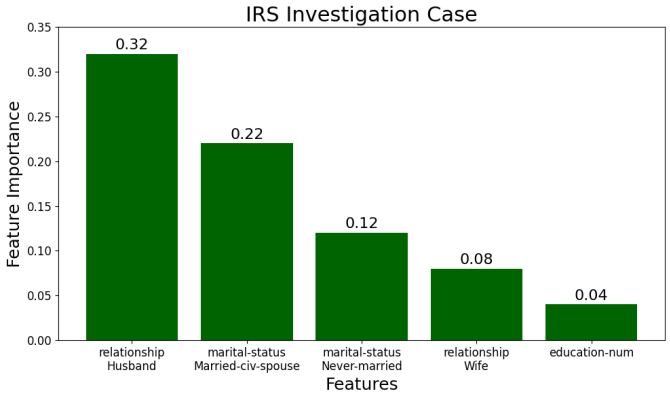

Fig. 9: IRS Fairness Histogram

In alignment with the EU AI Act's fairness and non discrimination requirements for high-risk systems, we excluded features that directly encode protected characteristics such as **race**, **sex**, and **native-country**. Additionally, we evaluated potential proxy variables like **marital-status** and **relationship**, and chose to remove them due to their strong association with demographic attributes. As a result of these fairness-driven exclusions, we observed a decrease in model performance: the **ROC AUC** dropped by approximately **5–6%**, which is expected given the removal of high-importance but potentially biased features. Other performance metrics showed smaller declines, typically around **1%**.

The updated top five features and their importance scores for each use case are listed below:

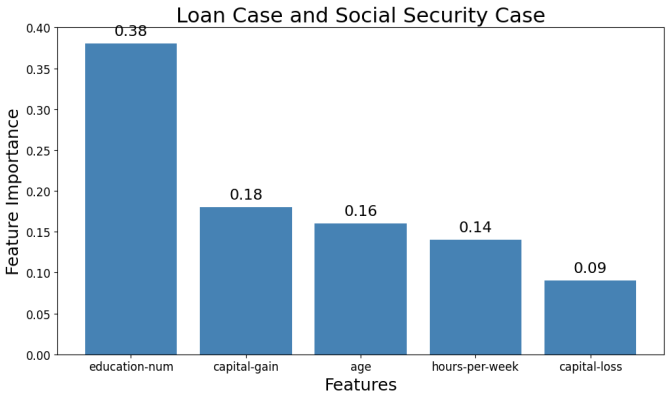

Fig. 10: Loan and Social Security Fairness Histogram after removal

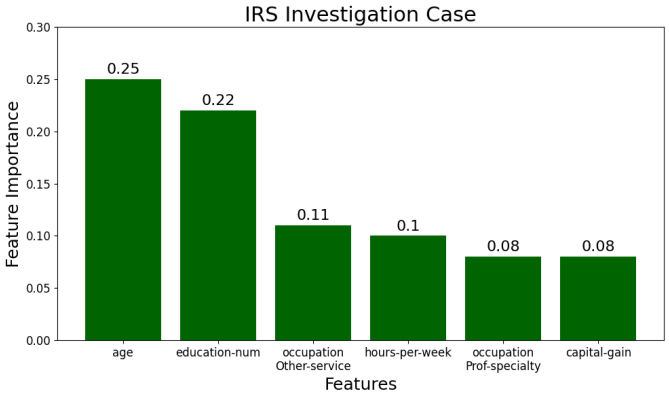

Fig. 11: IRS Fairness Histogram after removal

CONCLUSION

This project explored the application of machine learning models to predict individual income levels using the Adult Census dataset, with a focus on three real-world high-impact use cases: high-risk loan approval, IRS auditing, and social security benefits eligibility. We implemented thorough preprocessing, applied sampling strategies, and tuned models specifically for each scenario.

Among the evaluated models, XGBoost consistently outperformed others across most metrics, demonstrating strong predictive capacity even under varying data splits and resampling strategies. Hyperparameter tuning yielded great performance improvements aligned with each use case's goals, although often at the expense of trade-offs in other metrics.

In accordance with the EU AI Act's fairness guidelines, we removed sensitive features such as race, sex, native-country, marital-status, and relationship, to mitigate potential bias and ensure compliance with non-discrimination principles.

Overall, this study shows that machine learning can be responsibly applied to socio-economic classification tasks, offering real-world utility while being fair and ethical. We really enjoyed working on this project and learned a lot about the practical and ethical challenges of applying machine learning to real-world socio-economic problems, as well as how to put to practice the theoretical contents we learned in class.

ACKNOWLEDGMENTS

It is important to note that although most of this report was written by hand, it was done with the help of AI tools, mostly in the **State of the Art** section where a more technical approach was needed. The tools used were **Consensus** (https://consensus.app/) as recommended by the course's teacher, which helped with research while providing trustworthy papers as a source, this also helped with the fact that a lot of these papers are not public and free to read. **Claude-3.7-Sonnet** by **Anthropic** was also used for providing starting points on what to write, as well as what topics to search. For developing the models and code in general, AI was also used, but never to reach any conclusions considering what models to use. Finally, **Claude** was again used to review most of the text, helping with technical correctness.

REFERENCES

[1] Matkowski, M., Edinaldo, A. & Nguyen, S. Prediction of Individual Level Income: A Machine Learning Approach. (2022)

[2] Mullainathan, S. & Spiess, J. Machine Learning: An Applied Econometric Approach. *Journal Of Economic Perspectives*. **31**, 87-106 (2017)

[3] Asaduzzaman, A., Uddin, M. R., Woldeyes, Y. & Sibai, F. N. A Novel Salary Prediction System Using Machine Learning Techniques. *2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference*. (2024)

[4] Wang, J. Research on Income Forecasting based on Machine Learning Methods and the Importance of Features. *Proceedings of the International Conference on Information Economy, Data Modeling and Cloud Computing*. (2022)

[5] Lazar, A. Income prediction via support vector machine. *2004 International Conference on Machine Learning and Applications*. (2004)

[6] Ellum, R., Lewis, A., Xhaferaj, K., Shipsey, R., Račinskij, V. & White, Z. Machine Learning for Data Linkage. *International Journal of Population Data Science*. **8(2)** (2023)

[7] Echevin, D., Fotso, G., Bouroubi, Y., Coulombe, H. & Li, Q. Combining survey and census data for improved poverty prediction using semi-supervised deep learning. *Journal of Development Economics*. (2024)

[8] Zhang, B., Lemoine, B. & Mitchell, M. Mitigating Unwanted Biases with Adversarial Learning. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. (2018)

[9] Chockalingam, V., Shah, S. & Shaw, R. Income Classification using Adult Census Data (CSE 258 Assignment 2). (2017)

[10] XGBoost Documentation. https://xgboost.readthedocs.io/en/stable/. Accessed 15/05/2025

[11] AI Act Recital 27. https://artificialintelligenceact.eu/recital/27/#:~:text=Diversity%2C%20non%2Ddiscrimination%20and%20fairness%20means%20that%20AI%20systems%20are,by%20Union%20or%20national%20law. Accessed 29/05/2025.