Thuy Nguyen, Sumi Nguyen, Jayden Tran

COMPSCI 178

Professor Berg

March 18th, 2025

Analysis of Adults Data: Income Prediction

1. Introduction

In this project, we analyze the Adult dataset, a collection of demographic and employment-related information. The primary task is to predict an individual's income level in the United States, specifically whether they earn more than $50,000 annually or less. This task, known as income classification, has significant applications in various fields such as marketing, economics, and social sciences, where understanding income distribution is essential for decision-making. Given the complexity of the dataset, which includes both numerical and categorical features such as age, education level, occupation, and capital gains, we aim to explore which machine learning model can best classify income levels. In particular, we seek to determine which model is most effective at distinguishing between individuals with incomes above and below the $50,000 threshold. Moreover, we are also interested in identifying which features contribute most significantly to income prediction. By analyzing factors like education, age, and work class, we can uncover key drivers that influence an individual's income. To achieve this, we will experiment with several models, including Binary Logistic Regression, SGDC, Random Forest, K-Nearest Neighbor (KNN), and Extreme Gradient Boosting (XGB), and evaluate their performance to find the best approach for this classification task.

2. Methodology

a. About dataset

The dataset used in this project is the Adult dataset, also known as the Census Income dataset. It consists of personal and demographic information collected from individuals, with attributes detailed in the original dataset link.

b. Data Preprocessing

As the first step of our data cleaning process, we dropped all rows (representing each data point) that contained missing values. Additionally, since our focus is on individuals in the U.S., we selected only the data points where the value of 'country' is equal to 'United-States.' We observed 15 features in total, but not all of them contain meaningful data. In particular, the columns 'capital-loss' and 'capital-gain' were mostly filled with zeros. To confirm our observations, we visualized the distribution of our data into these two columns.

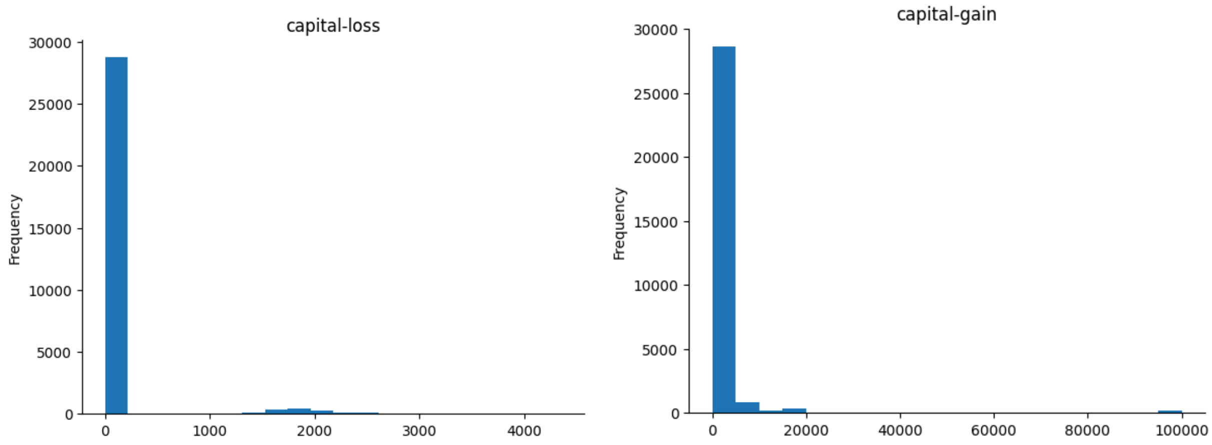
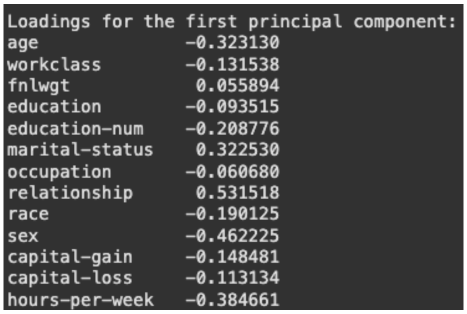


Figure 1. Frequency distribution of capital-loss and capital-gain

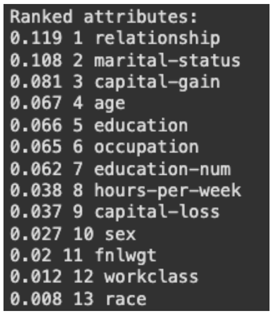
Upon examining the distributions of capital-gain and capital-loss, we noticed that a significant number of values were zero, which aligns with our initial observation. This suggests that these features have limited variability and may not contribute significantly to distinguishing between income classes. However, before making a final decision, we want to ensure that these features are not significant in our dataset by using Principle Component Analysis (PCA).



```
Loadings for the first principal component:
age              -0.323130
workclass        -0.131538
fnlwgt            0.055894
education        -0.093515
education-num    -0.208776
marital-status    0.322530
occupation       -0.060680
relationship      0.531518
race             -0.190125
sex              -0.462225
capital-gain     -0.148481
capital-loss     -0.113134
hours-per-week   -0.384661
```

Figure 2. PCA Loadings for the First Principal Component

Based on the PCA loadings, we analyzed the contribution of each feature to the principal components. Features with weak or insignificant loadings below 0.1, such as fnlwgt, education, and occupation, were deemed unnecessary in explaining the variance captured by the principal components. Nevertheless, among the three features, we believe that education and occupation will have a particular impact on an individual's income. Therefore, we decided to conduct another attribute selector, the Significance Attribute Evaluator (SAE). The result, as shown in Figure 3, suggests that education and occupation have significant classification factors.



```
Ranked attributes:
0.119 1 relationship
0.108 2 marital-status
0.081 3 capital-gain
0.067 4 age
0.066 5 education
0.065 6 occupation
0.062 7 education-num
0.038 8 hours-per-week
0.037 9 capital-loss
0.027 10 sex
0.02 11 fnlwgt
0.012 12 workclass
0.008 13 race
```

Figure 3. Ranked significance of features using SelectKBest with Mutual Information

Overall, through the results from PCA and SAE, we decided to drop the least significant feature indicated by both which is 'fnlwgt'.

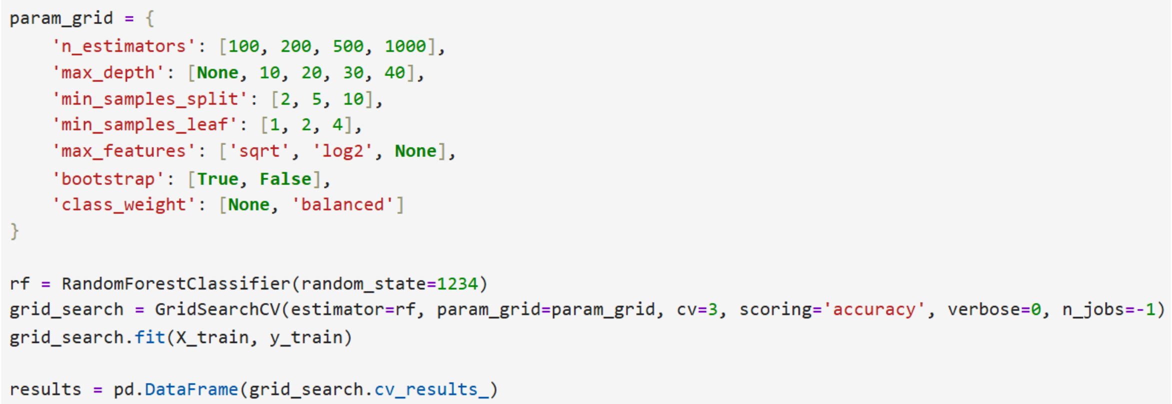
c. Model Training

We decided to use several machine learning models, including Binary Logistic Regression, Stochastic Gradient Descent (SGD) Classifier, Random Forest (RF) Classifier, Classifier K-Nearest Neighbor (KNN) Classifier, Extreme Gradient Boosting (XGBoost), and

Support Vector Machine (SVM), to process data efficiently compared to more complex alternatives. Following this, we performed parameter tuning for each model.

For the KNN Classifier, which has one parameter, `n_neighbors`, we conducted a brute-force search to determine the number of neighbors and evaluate the performance.

For other models, including Binary Logistic Classification, SGD Classifier, RF Classifier, SVM, and XGBoost, which have more parameters that significantly affect the results of the models, we utilized GridSearchCV as our parameter tuner. We examined all possible values for each parameter based on our dataset's characteristics. This approach allowed us to find the best parameters for each model using a subset of our data. As a result, we identified the most optimal parameters for each model. However, the SVM model was excluded from the figure due to its inefficiency resulting from the high training time.



```
param_grid = {
    'n_estimators': [100, 200, 500, 1000],
    'max_depth': [None, 10, 20, 30, 40],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2', None],
    'bootstrap': [True, False],
    'class_weight': [None, 'balanced']
}

rf = RandomForestClassifier(random_state=1234)
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3, scoring='accuracy', verbose=0, n_jobs=-1)
grid_search.fit(X_train, y_train)

results = pd.DataFrame(grid_search.cv_results_)
```

Figure 4: Random Forest model's parameters tuning process

3. Results

a. Best model

i. Training vs Testing

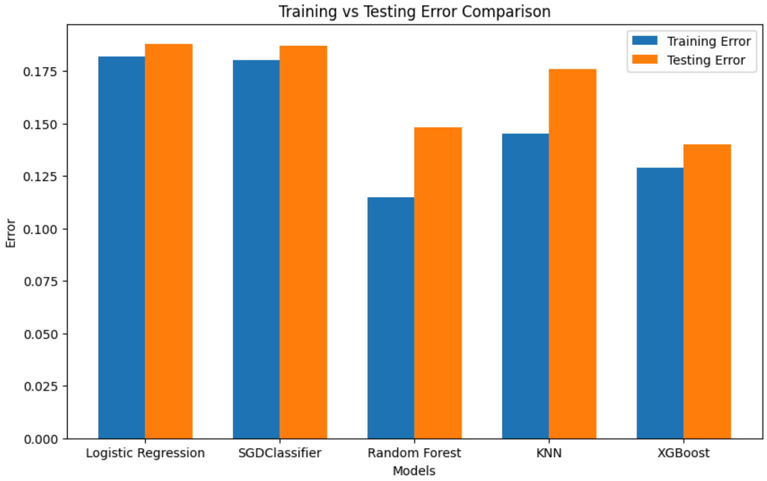


Figure 5. Training vs Testing Error across all models

After training all models, we visualize the training and testing error for all models to help decide the "best" among them. As Figure 5 suggests, although RF has the lowest training error, it has a relatively high testing error, indicating that the model might be overfitting. XGBoost has the second lowest training error and lowest testing error (these values are relatively close), and we claim that XGBoost is our best model for this classification problem. However, since our dataset is not balanced (as shown in Figure 6 below), we want to use ROC to help with class imbalance and make more reliable predictions.

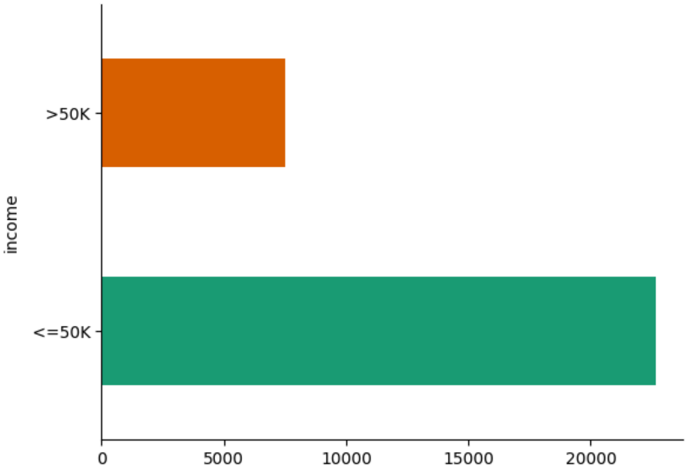


Figure 6. Frequency distribution of income (as a categorical variable)

ii. Receiver Operating Characteristic (ROC) Curve (Sumi)

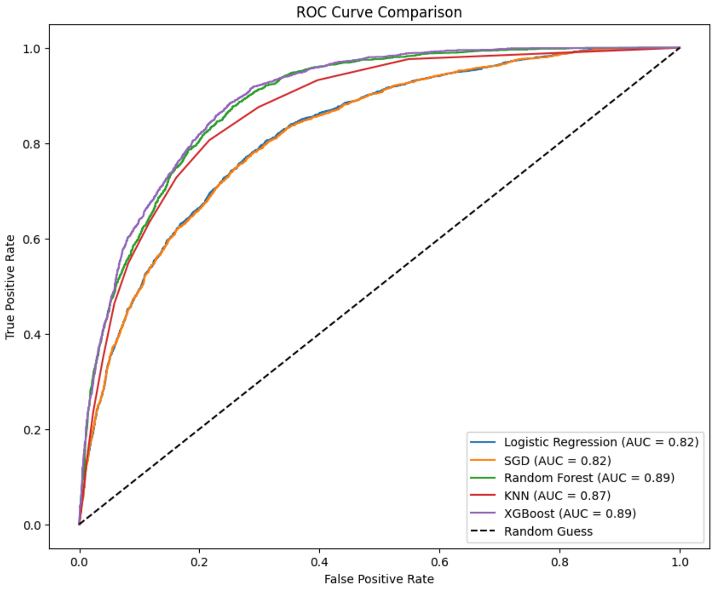


Figure 7. ROC Curve Comparison across all models

All models demonstrate significantly better performance than random guessing, each having an Area Under Curve (AUC) of over 0.80. Upon examination, both the XGBoost and RF models recorded the highest AUC of 0.89, closely followed by the KNN model with an AUC of 0.87.

iii. Conclusion

After evaluating both training and testing errors as well as ROC Curve evaluation, we concluded that the XGBoost model is the best choice for this problem, as its superior performance is highlighted by the lowest testing error and highest AUC.

b. Significant Feature Test

After choosing the most efficient model for the task, we focused on selecting features that would give the best training result for our XGBoost model. We trained the model with each feature dropped and recorded the training and testing error rates. Here are the error rates after dropping each of the 12 features:

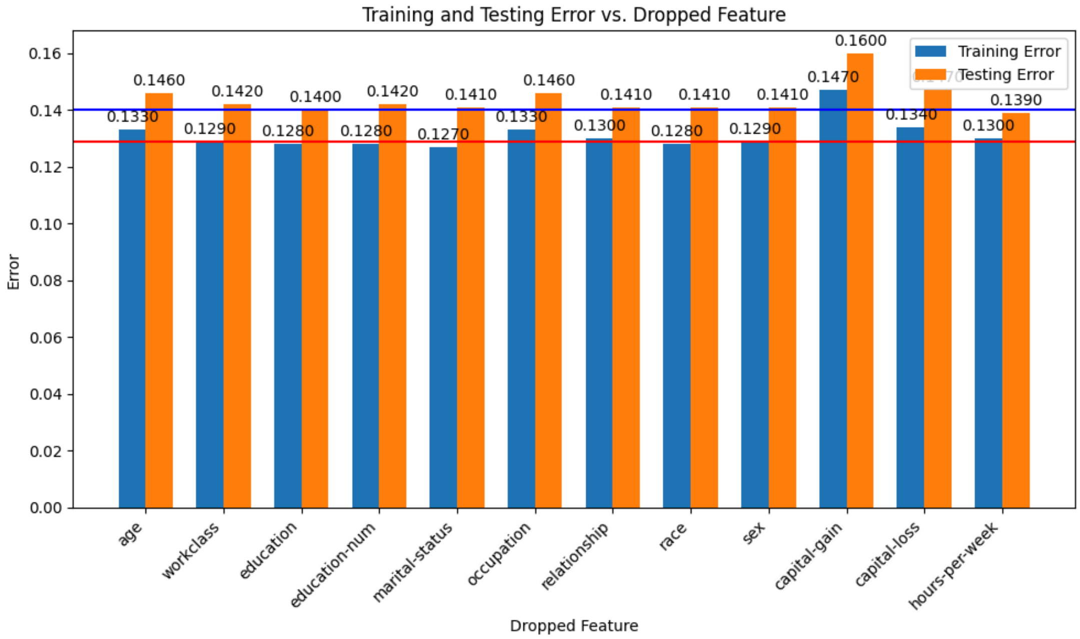


Figure 8: Comparison of Training and Testing Error Rates when a feature is dropped

We used the original dataset as the baseline, with red and blue lines representing the training and testing errors. For comparison, we considered any result with a ±0.002 difference compared to the baseline insignificant. After examining the table, we concluded that although removing the features “education,” “education-num,” “marital-status,” “race,” or “hours-per-week” would have a slight positive impact on the training error, they will either negatively affect the testing error or not be significant enough to consider.

4. Conclusion

We examined the effectiveness of various machine learning models for predicting income based on the Adult dataset. Our primary objective was to find a model that accurately classifies individuals as earning either more or less than $50,000 annually and to identify the key factors that influence a person’s income.

We began by preprocessing the dataset, addressing missing values, and analyzing feature relevance through PCA and SAE. This process led to the exclusion of the “fnlwgt” feature, which we deemed insignificant for our analysis. We then trained and evaluated the base dataset on several models, including Binary Logistic Classification, SGD Classifier, RF Classifier, KNN Classifier, and XGBoost, utilizing GridSearchCV for hyperparameter tuning to find the best-performing model. Ultimately, XGBoost demonstrated the best performance, with the lowest testing error and a high AUC value, indicating superior classification accuracy for income levels.

Furthermore, we conducted a significant feature test to refine our model. Although the removal of any single feature did not significantly enhance the model's performance, we gained a more profound understanding of each feature's influence on the predictions.

In conclusion, our study confirms the effectiveness of XGBoost for income classification on a dataset comprising diverse features. Notably, 12 out of 13 features of the Adult dataset are important in building the model.

5. Credit

- Thuy Nguyen
    - Report: Introduction, Data Preprocessing, Training vs Testing for best model
    - Model Training: Data Cleaning and preprocessing, Binary Logistic Regression, SGD Classifier
- Sumi Nguyen
    - Report: Model Training (KNN), ROC Curve, Conclusion for best model
    - Model Training: KNN, XGBoost, ROC Curve
- Jayden Tran
    - Report: Choosing Model, Significant Test, Overall Conclusion.
    - Model Training: Parameter tuning for Random Forest Classifier and SVM, Feature Selection After Choosing Model.

Code Reference:

https://colab.research.google.com/drive/1UMP7mqH0lUIjTiZyLFSJ8EshBYpyqlEt?usp=sharing