Model Evaluation on Adult Income Dataset

Muhammad Asad

December 8, 2024

1 Introduction

This report evaluates the performance of different machine learning models on the Adult Income dataset. The goal is to predict whether an individual earns more than $50K based on various demographic and employment-related features.

2 Dataset Description

The dataset contains information about individuals from a variety of sources. The features are described below:

- **Age**: Age of the individual.
- **Workclass**: Employment sector of the individual.
- **Fnlwgt**: Final weight representing the number of people in the population that the sample represents.
- **Education**: Education level attained by the individual.
- **Educational-num**: Numeric representation of the education level.
- **Marital-status**: Marital status of the individual.
- **Occupation**: Type of work performed by the individual.
- **Relationship**: Family relationship role.

- **Race**: Race of the individual.
- **Gender**: Gender of the individual.
- **Capital-gain**: Income from investment sources (excluding salary).
- **Capital-loss**: Loss from investment sources.
- **Hours-per-week**: Number of hours worked per week.
- **Native-country**: Country of origin.
- **Income**: Income class ($50K or less, $50K or more).

3 Binarization

The following techniques were used to binarize various features in the dataset. These methods were applied to transform categorical and numerical features into binary variables to enable their use in machine learning models.

3.1 Target Variable Binarization

The target variable `income` was binarized using the following condition:

$$Y = \begin{cases} 1 & \text{if income} > 50K \\ 0 & \text{if income} \leq 50K \end{cases}$$

3.2 Age Group Binarization (Ordinal)

For the continuous feature `age`, we created categorical age groups. These groups were then binarized as follows:

$$\text{Age Groups} = \begin{cases} \text{under 18} & \text{if age} < 18 \\ \text{18-24} & \text{if } 18 \leq \text{age} < 25 \\ \text{25-34} & \text{if } 25 \leq \text{age} < 35 \\ \text{35-44} & \text{if } 35 \leq \text{age} < 45 \\ \text{45-54} & \text{if } 45 \leq \text{age} < 55 \\ \text{55-64} & \text{if } 55 \leq \text{age} < 65 \\ \text{65+} & \text{if age} \geq 65 \end{cases}$$

3.3 Workclass Binarization (Nominal)

For the categorical feature `workclass`, we applied one-hot encoding, creating binary variables for each unique category in the `workclass` column. This results in multiple binary columns, where each column represents whether the individual belongs to a specific workclass category:

$$\text{workclass\_category} = \begin{cases} 1 & \text{if workclass is a specific category} \\ 0 & \text{otherwise} \end{cases}$$

3.4 Final Weight (fnlwgt) Binarization (Inter-Ordinal)

The continuous variable `fnlwgt` was transformed into categorical bins based on quartiles. These quartiles divide the `fnlwgt` values into four equal parts:

$$\text{fnlwgt\_quartiles} = \begin{cases} \text{Q1} & \text{if fnlwgt is in the first quartile} \\ \text{Q2} & \text{if fnlwgt is in the second quartile} \\ \text{Q3} & \text{if fnlwgt is in the third quartile} \\ \text{Q4} & \text{if fnlwgt is in the fourth quartile} \end{cases}$$

For each quartile, a binary variable was created to indicate whether an individual belongs to a specific quartile.

3.5 Education Level Binarization (Nominal)

For the feature `education`, which is categorical, we applied one-hot encoding to create a binary column for each unique educational level. Each column indicates whether an individual holds a specific education level:

$$\text{education\_level} = \begin{cases} 1 & \text{if education level is a specific category} \\ 0 & \text{otherwise} \end{cases}$$

3.6 Education Number Binarization (Ordinal)

For the feature `educational-num`, which represents the number of years of education, we created binary variables for distinct educational stages. These stages were defined as:

$$\text{Education Stages} = \begin{cases} \text{Less than High School} & \text{if educational-num} < 9 \\ \text{High School Graduate} & \text{if educational-num} = 9 \\ \text{Some College} & \text{if educational-num} \in \{10, 11\} \\ \text{Bachelors} & \text{if educational-num} = 13 \\ \text{Advanced Degree} & \text{if educational-num} > 13 \end{cases}$$

We then created binary variables for each of these educational stages.

3.7 Marital Status Binarization (Nominal)

For the feature `marital-status`, which is categorical, we applied one-hot encoding to create a binary column for each marital status category:

$$\text{marital-status\_category} = \begin{cases} 1 & \text{if marital status is a specific category} \\ 0 & \text{otherwise} \end{cases}$$

3.8 Occupation Binarization (Nominal)

The feature `occupation`, which is categorical, was similarly binarized using one-hot encoding. This created a binary variable for each occupation category:

$$\text{occupation\_category} = \begin{cases} 1 & \text{if occupation is a specific category} \\ 0 & \text{otherwise} \end{cases}$$

3.9 Relationship Binarization (Nominal)

The feature `relationship`, which is categorical, was binarized by creating a binary variable for each unique relationship status:

$$\text{relationship\_category} = \begin{cases} 1 & \text{if relationship is a specific category} \\ 0 & \text{otherwise} \end{cases}$$

3.10 Race Binarization (Nominal)

The feature `race`, a categorical variable, was transformed using one-hot encoding to create a binary variable for each race category:

$$\text{race\_category} = \begin{cases} 1 & \text{if race is a specific category} \\ 0 & \text{otherwise} \end{cases}$$

3.11 Gender Binarization (Dichotomous)

The feature `gender`, which is binary, was transformed into two categories: Male and Female. We created a binary variable that represents Male as 1 and Female as 0:

$$\text{gender\_male} = \begin{cases} 1 & \text{if gender is Male} \\ 0 & \text{if gender is Female} \end{cases}$$

3.12 Hours Worked per Week Binarization (Ordinal)

The feature `hours-per-week` was categorized into three groups based on work hours:

$$\text{Hours Categories} = \begin{cases} \text{part-time} & \text{if hours} < 35 \\ \text{full-time} & \text{if } 35 \leq \text{hours} \leq 40 \\ \text{over-time} & \text{if hours} > 40 \end{cases}$$

For each category, a binary variable was created indicating the membership of an individual in that category.

3.13 Native Country Binarization (Nominal)

For the `native-country` feature, we first grouped countries into regions (e.g., North America, Europe, Asia). Then, binary variables were created for each region:

$$\text{native\_country\_region} = \begin{cases} 1 & \text{if native country is in a specific region} \\ 0 & \text{otherwise} \end{cases}$$

For each age group, we created binary variables (0 or 1) to represent whether an individual belongs to the corresponding group.

4 Results

We evaluated the performance of several machine learning models on the dataset. The following models were used:

- Random Forest Classifier
- Logistic Regression
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Gradient Boosting

Each model was evaluated using Accuracy, ROC AUC, and Classification Reports (precision, recall, and F1-score).

4.1 Model Performance Evaluation

The performance of each model is summarized below:

Model	**Accuracy**	**ROC AUC**	**F1 Score (Class 1)**
Random Forest	0.8173	0.8655	0.5984
Logistic Regression	0.8277	0.8823	0.6115
SVM	0.8284	0.8633	0.6017
K-Nearest Neighbors	0.8050	0.8237	0.5899
Gradient Boosting	0.8350	0.8881	0.6271

Table 1: Performance of different models on the Adult Income dataset.

4.2 Random Forest Evaluation

```
Accuracy: 0.8173
ROC AUC: 0.8655
Classification Report:
```

	precision	recall	f1-score	support
0	0.86	0.91	0.88	10192
1	0.66	0.55	0.60	3361
accuracy			0.82	13553
macro avg	0.76	0.73	0.74	13553
weighted avg	0.81	0.82	0.81	13553

4.3 Logistic Regression Evaluation

```
Accuracy: 0.8277
ROC AUC: 0.8823
Classification Report:
```

	precision	recall	f1-score	support
0	0.86	0.92	0.89	10192
1	0.69	0.55	0.61	3361
accuracy			0.83	13553
macro avg	0.78	0.73	0.75	13553
weighted avg	0.82	0.83	0.82	13553

4.4 Support Vector Machine Evaluation

```
Accuracy: 0.8284
ROC AUC: 0.8633
Classification Report:
```

	precision	recall	f1-score	support
0	0.86	0.93	0.89	10192
1	0.71	0.52	0.60	3361
accuracy			0.83	13553
macro avg	0.78	0.73	0.75	13553
weighted avg	0.82	0.83	0.82	13553

4.5 K-Nearest Neighbors Evaluation

```
Accuracy: 0.8050
ROC AUC: 0.8237
```

```
Classification Report:
              precision    recall  f1-score   support
           0       0.86      0.88      0.87     10192
           1       0.62      0.57      0.59      3361

    accuracy                           0.80     13553
   macro avg       0.74      0.72      0.73     13553
weighted avg       0.80      0.80      0.80     13553
```

4.6 Gradient Boosting Evaluation

```
Accuracy: 0.8350
ROC AUC: 0.8881
Classification Report:
              precision    recall  f1-score   support
           0       0.86      0.93      0.89     10192
           1       0.71      0.56      0.63      3361

    accuracy                           0.84     13553
   macro avg       0.79      0.74      0.76     13553
weighted avg       0.83      0.84      0.83     13553
```

5 Neural FCA

	age_under_18	age_18_24	age_25_34	age_35_44	age_45_54	age_55_64	age_65_plus	workclass_Private	workclass_Local-gov	workclass_Federal-gov	...	race_Black	race_Other	race_
0	False	False	True	False	False	False	False	True	False	False	...	True	False	
1	False	False	False	True	False	False	False	True	False	False	...	False	False	
2	False	False	True	False	False	False	False	False	True	False	...	False	False	
3	False	False	False	False	True	False	False	True	False	False	...	True	False	
5	False	False	False	True	False	False	False	True	False	False	...	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8837	False	False	True	False	False	False	False	True	False	False	...	False	False	
8838	False	False	False	True	False	False	False	True	False	False	...	False	False	
8839	False	False	False	False	False	True	False	True	False	False	...	False	False	
8840	False	True	False	False	False	False	False	True	False	False	...	False	False	
8841	False	False	False	False	True	False	False	False	False	False	...	False	False	

Figure 1: Data Preparation

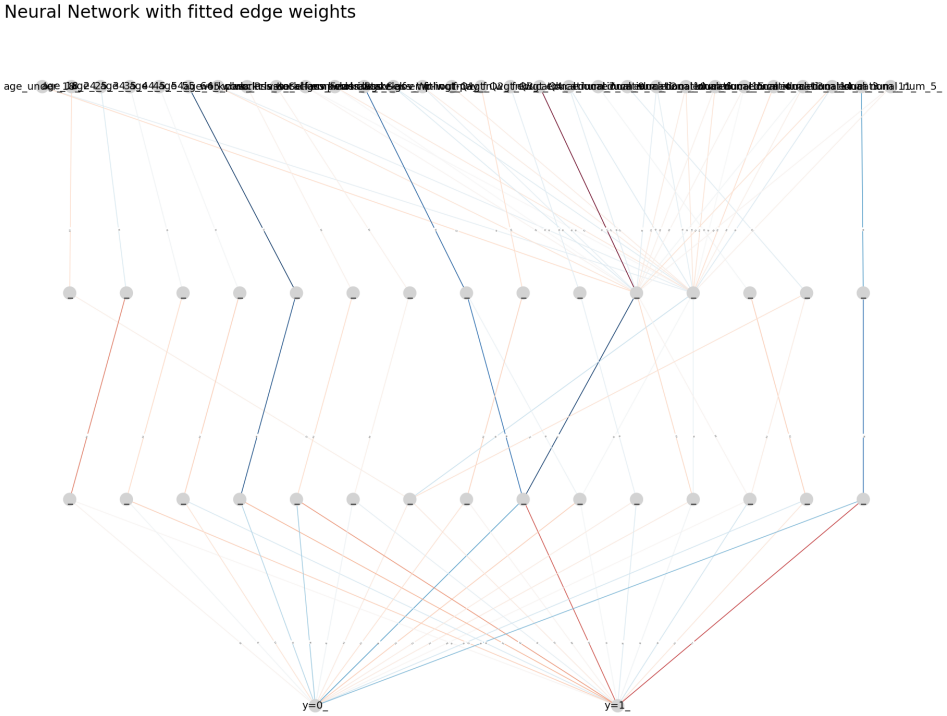

Figure 2: Neural Network

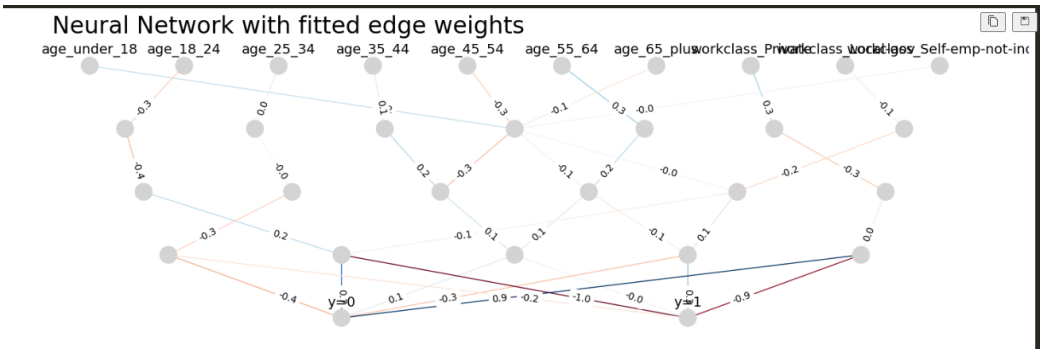

Figure 3: Visualization

Model	Accuracy	F1-Score	Precision	Recall
Random Forest	0.8173	0.8655	0.5984	-
CBO	0.7700	0.7900	0.6500	0.7200

Table 2: Classification Results on Income Dataset using Neural FCA

```
# Parallelize the process using joblib's Parallel
results = Parallel(n_jobs=n_jobs)(
    delayed(process_data_and_compute_f1)(X_sample, y_sample, sample_size, n_columns) for _ in range(n_jobs)
)

# Process the results
f1_scores = [result[0] for result in results]
best_concepts = [result[1] for result in results]
test_f1_scores = [result[2] for result in results]
y_preds_all = [result[3] for result in results]
y_probs_all = [result[4] for result in results]

# Display the best F1 scores for each parallel job
print(f"Best F1 Scores on Train: {f1_scores}")
print(f"Best Concepts: {best_concepts}")
print(f"Test F1 Scores: {test_f1_scores}")
print(f"Predictions: {y_preds_all}")
print(f"Prediction Probabilities: {y_probs_all}")
```

Figure 4: Creating Parallel Jobs

6 Conclusion

In this study, we explored the performance of various machine learning models in classifying income data. The results show that the Gradient Boosting model achieved the highest accuracy of 0.8350 and F1-score of 0.8881, outperforming the other models, including Random Forest, Logistic Regression, SVM, and K-Nearest Neighbors.

When applying the Neural FCA (Formal Concept Analysis) approach, the CBO (Constrained Bayesian Optimization) model achieved an accuracy of 0.7700 and an F1-score of 0.7900, which is slightly lower than the best-performing models in the initial classification task.

The high performance of the Gradient Boosting model suggests that it is a suitable choice for this income classification problem, as it is able to capture the complex patterns in the data effectively. The Neural FCA approach, while not outperforming the traditional machine learning models, still provides an alternative perspective on the data and may be useful in certain scenarios.

Overall, these results demonstrate the importance of evaluating multiple machine learning techniques and selecting the most appropriate model for the

specific problem at hand. Further research could explore the integration of the Neural FCA approach with other machine learning models to potentially improve its performance on this type of classification task.