Predicting Income Level from Census Data

Team: Yuyang Peng, Alan Zhang, Gevin Yang

Introduction:

Our project aims to utilize the Census Income dataset, extracted from the 1994 Census database by Barry Becker, as a valuable resource for social science research and predictive modeling tasks. With 48,842 instances and 14 features, this dataset offers a comprehensive look into various demographic and socioeconomic factors. Our goal is to predict whether an individual's income exceeds $50,000 per year based on these factors.

The Census Income dataset is multivariate and primarily intended for classification tasks. Each instance represents an individual and includes features such as age, workclass, education level, marital status, occupation, race, sex, capital gains, capital losses, hours worked per week, and native country. The target variable, 'income,' is categorized as '>50K' or '<=50K', indicating whether an individual's income surpasses $50,000 annually.

By leveraging this rich dataset, we aim to explore the intricate relationships between demographic attributes and income levels. Through predictive modeling, we seek to uncover patterns and insights that can inform policy decisions, social interventions, and economic strategies aimed at promoting financial equity and opportunity.

Exploratory Analysis:

Statistically Descriptive Analysis of The Dataset

The Census Income dataset comprises 48,842 instances with 14 features. The dataset is multivariate and primarily aimed at classification tasks. Each instance represents an individual and includes features such as age, workclass, education level, marital status, occupation, race, sex, capital gains, capital losses, hours worked per week, and native country. The target variable is 'income,' categorized as '>50K' or '<=50K', representing whether an individual's income exceeds $50,000 annually as it shown in Fig 1.

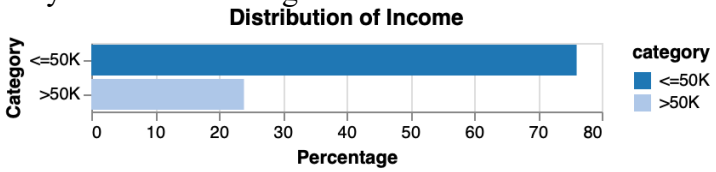


Fig 1

In our correlation map (Fig 2), we aimed to discern the relationship between independent features and the dependent variable. Observably, most features exhibit a positive correlation with the income variable.

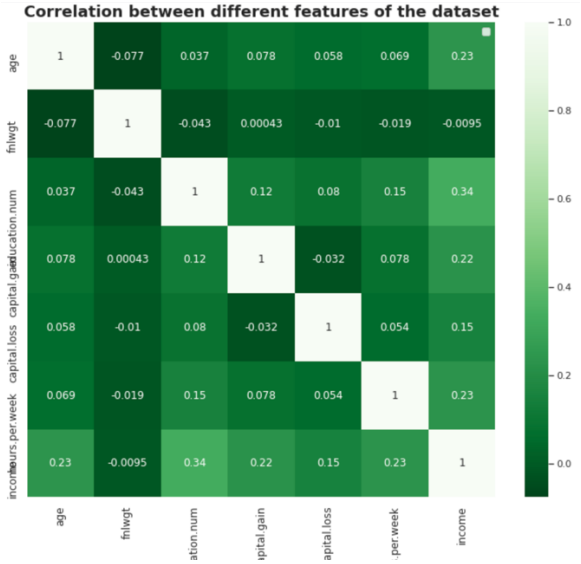


Fig 2

Detailed Description for Selected Features

This dataset involves 14 variables, and 2 out of 14 are continuous and the others are categorial. The exploratory data analysis below is designed for four selected features including the distribution of both categorial and continuous variables distinguished by the two groups of the target features.

Feature 1: The 'age' feature in the dataset represents individuals' ages, encoded as integer values ranging from 17 to 90. There are no missing values present in this feature. By plotting the density graph, we find that gamma distribution fits it well.

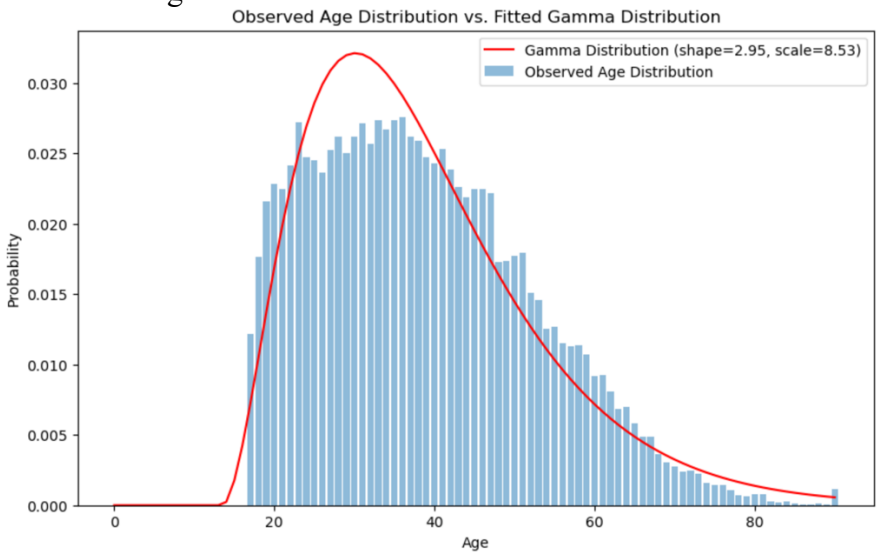


Fig 3

Feature 2: The 'workclass' feature is categorical, with missing values.

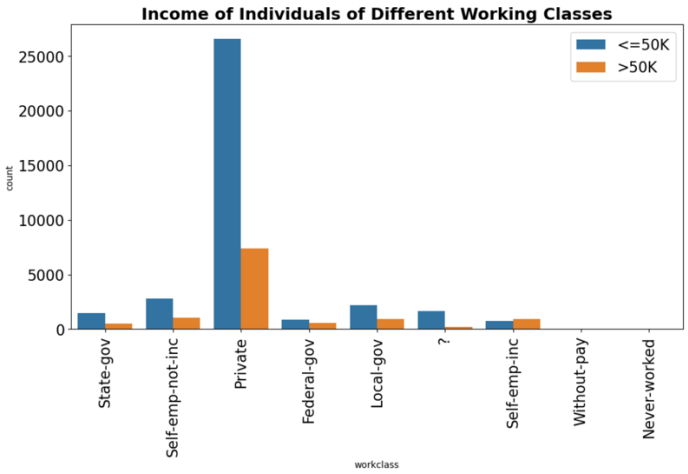


Fig 4

The graph categorizes adult incomes by employment sector. We can see self-Employed individuals earning over 50K dollars outnumber those earning less. Private Sector employees see a significant income gap, with most earning under 50K. There's little disparity in income for Federal Government employees. Data for those Without-Pay or Never-Worked is sparse.

Feature 3: The 'race' attribute within the dataset is of categorical type, encompassing diverse racial classifications with no instances of missing values.

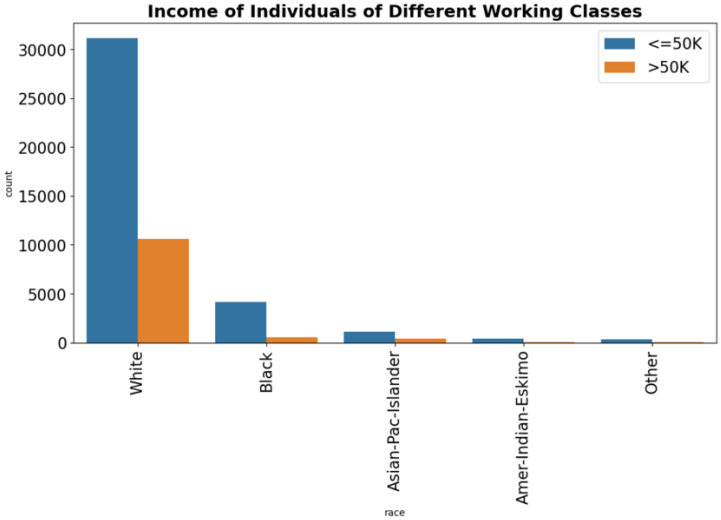


Fig 5

It's noteworthy that, apart from Whites, there are relatively few individuals from other racial groups in the dataset. As a result, it might be challenging to accurately assess the percentage and correlation of individuals earning over 50K dollars annually.

Feature 4: The "relationship" feature consists of categorical data representing various relationship statuses, such as Wife, Own-child, Husband, Not-in-family, Other-relative, and Unmarried, with no missing values.

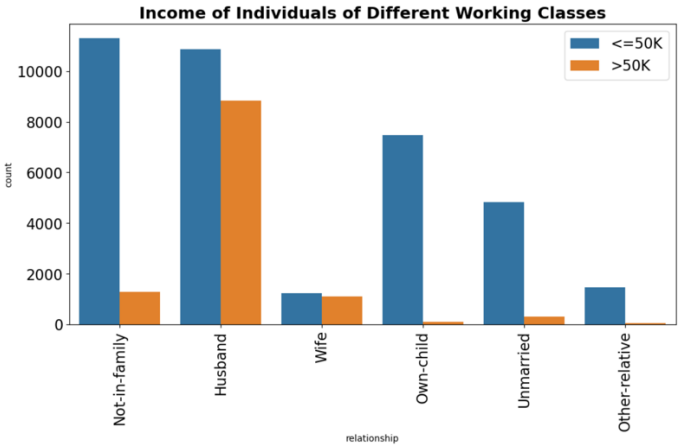


Fig 6

Wives show an equal likelihood of earning more than 50K dollars per year compared to husbands. However, husbands exhibit a slightly lower probability of earning above 50K annually. Additionally, there are very few unmarried individuals who earn more than 50K dollars annually.

This summary provides a comprehensive overview of each feature in the dataset, including data type, range, categories (if categorical), and presence of missing values.

Methodology:

Data Cleaning

For the raw data, we could find there are four unique for target features (shown in Fig 7).

```
y['income'].unique()
✓ 0.0s
array(['<=50K', '>50K', '<=50K.', '>50K.'], dtype=object)
```

Fig 7

But the labels of “<=50K” and “<=50K.” should be classified to the same group and the same to the other two labels. We replaced the abnormal labels resulting in two final values for target features: “<=50K” and “>50K”.

Summary of the dataset shows that there are no missing values. But the preview shows that the dataset contains values coded as `?` in “work class”, “occupation” and “native country”. Then we replaced those characters with NaN values.

To ensure the integrity and completeness of our analysis, it is imperative to address missing values within the dataset. In this regard, we have opted to utilize the mode imputation method to fill in any NaN (Not a Number) values present in three specific features. This method entails replacing missing values with the most frequently occurring value within each respective feature.

Data Splitting

Because the number of this dataset is not small, it is not that necessary to utilize cross-validation for training and test. The dataset is applied by "7-3" splitting randomly into training data and test data.

Introduction to Principal Component Analysis

Principal component analysis, or PCA, is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

From the heatmap (Fig 2) we can find that, there is certain collinearity between some variables, so applying PCA to this dataset, and comparing results with/without this technique seems a reasonable idea.

To apply PCA and implement models to the dataset, converting the categorial values to numeric is necessary. However, it is crucial to note that PCA is sensitive to the scale of features within the dataset. Inconsistencies in feature scales can lead to skewed principal components and potentially biased results.

To address this sensitivity, it is essential to preprocess the data by scaling the features appropriately before the following work. Scaling ensures that all features contribute equally to the variance calculation and the subsequent generation of principal components.

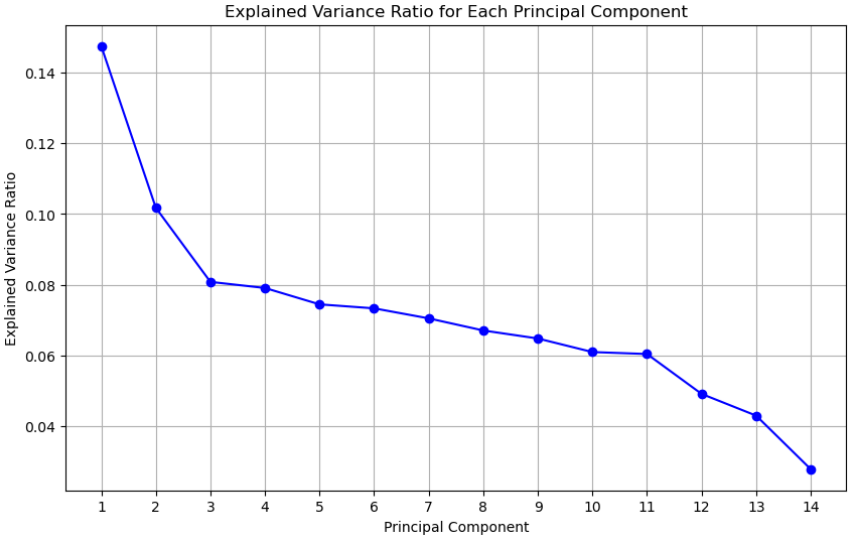


Fig 8

As Fig 8 shows, the first two PCs explains much more than other PCs.

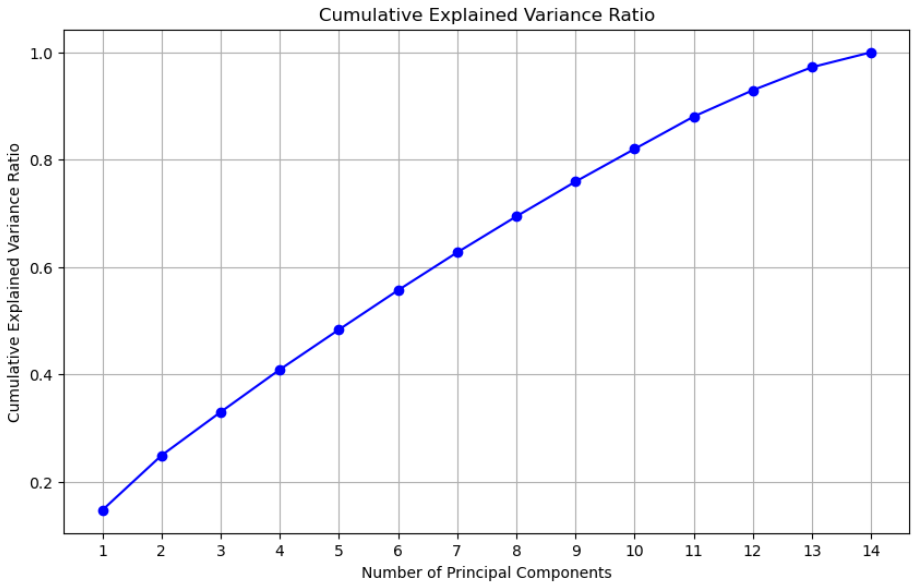


Fig 9

After conducting PCA on the dataset, it was found that the top 12 principal components (PCs) collectively explain approximately 90% of the variance in the dataset. This indicates that a significant portion of the variability present in the original features can be captured by these principal components.

Logistic Regression

Logistic regression is a fundamental statistical technique used for binary classification tasks, where the target variable is categorical with two possible outcomes. Unlike linear regression, which predicts continuous values, logistic regression predicts the probability of an observation belonging to a particular class.

In the initial phase, logistic regression was employed to analyze the entire set of features. Following this, logistic regression analysis was extended to focus solely on the first 12 principal components.

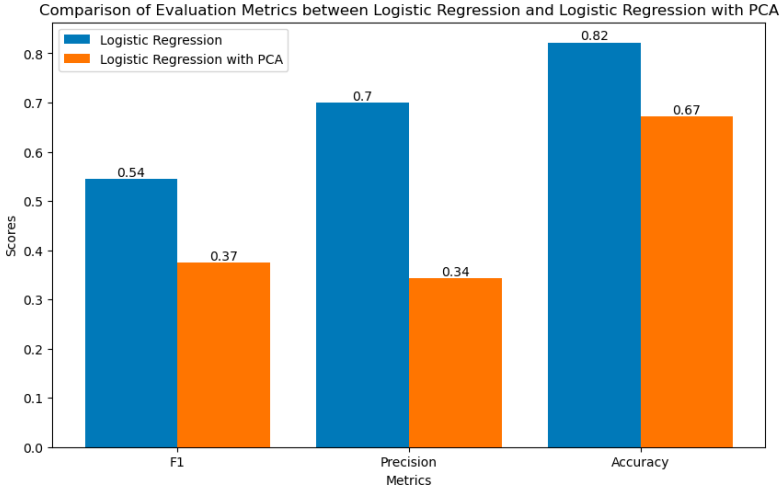


Fig 10

From the graph, we can observe that the results of PCA applied to logistic regression consistently show lower F1 scores, precision, and accuracy compared to using all features. This phenomenon can be explained by several factors:

Information Loss during Dimensionality Reduction: PCA aims to capture the maximum variance in the data with a reduced number of components. However, in the process of dimensionality reduction, some information from the original features may be lost. This loss of information can result in a less discriminative representation of the data, leading to inferior model performance.

Loss of Feature Specificity: While principal components retain some information from the original features, they are linear combinations of these features. Consequently, they may lack the specificity and interpretability of the original features, making it more challenging for the model to accurately capture the underlying patterns in the data.

Generalized Addictive Model (GAM)

Generalized Additive Models (GAMs) stand out in the realm of statistical modeling, bridging the gap between the interpretability of linear models and the flexibility of non-linear models. They excel in binary classification tasks, where the outcome variable is dichotomous. (Just like in this case where we have 2 types of response variables.) GAMs maintain the core concept of logistic regression, predicting probabilities of class membership, but with an added twist: they allow for the incorporation of non-linear relationships between predictors and the log odds of the outcome.

To run the GAM model, we first must transform all the categorical variables into dummy variables, the dataset now comes to 29305 rows × 108 columns.

For each of the numerical and categorical variables, we use s() to specify smooth terms in a GAM, using splines for non-linear relationships between a predictor and the outcome, enhancing the model's capacity to capture data complexity. We use f() to define factor terms for categorical variables in a GAM, fitting separate constants for each category, similar to employing dummy variables, allowing the model to account for each category's unique impact.

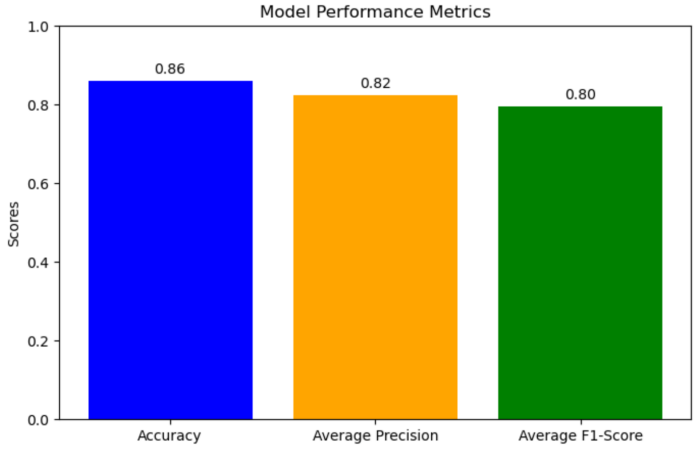


Fig 11

For the graph, the model correctly predicts the outcome 86% of the time. High accuracy indicates good overall performance but doesn't reflect the balance between classes or how the model performs in each class. On average, when the model predicts an instance as positive, it is correct 82% of the time. This suggests the model has a lower tendency to incorrectly label negative instances as positive. The F1 score combines precision and recall into a single metric, accounting for both false positives and false negatives. An average F1-score of 0.80 indicates a good balance between precision and recall, meaning the model reasonably predicts positive instances and doesn't miss out on too many actual positive instances.

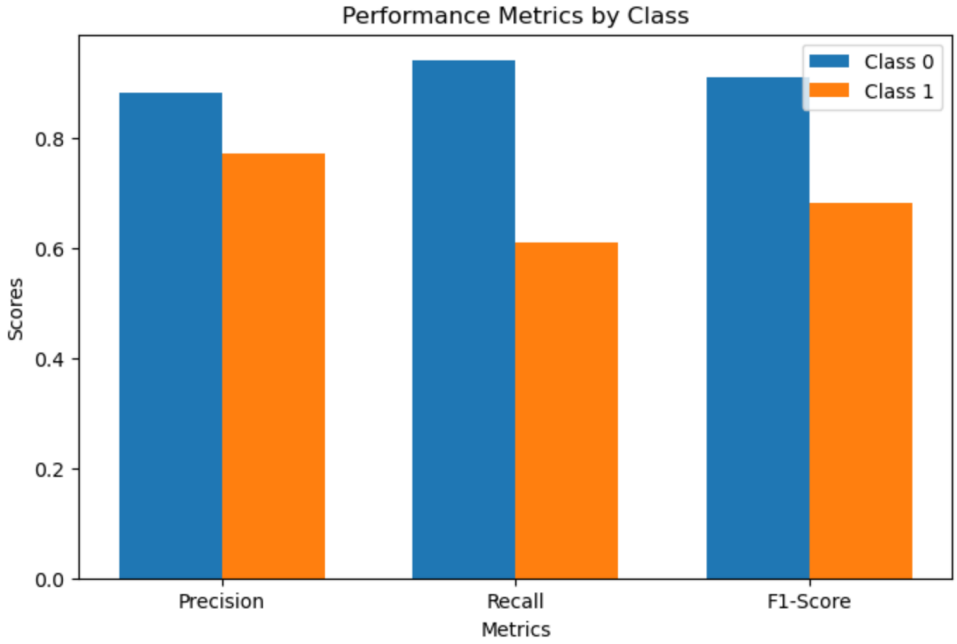


Fig 12

Overall, the model seems to perform better when predicting Class 0 outcomes. This could be indicative of a class imbalance or that Class 1 is more challenging to predict correctly due to various factors such as overlapping feature values or insufficient representation in the training data.

Random Forest

Random Forest is a versatile machine-learning method capable of performing both regression and classification tasks. For binary classification, where the target variable has two categories, Random Forest is particularly effective due to its ensemble approach. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes predicted by individual trees.

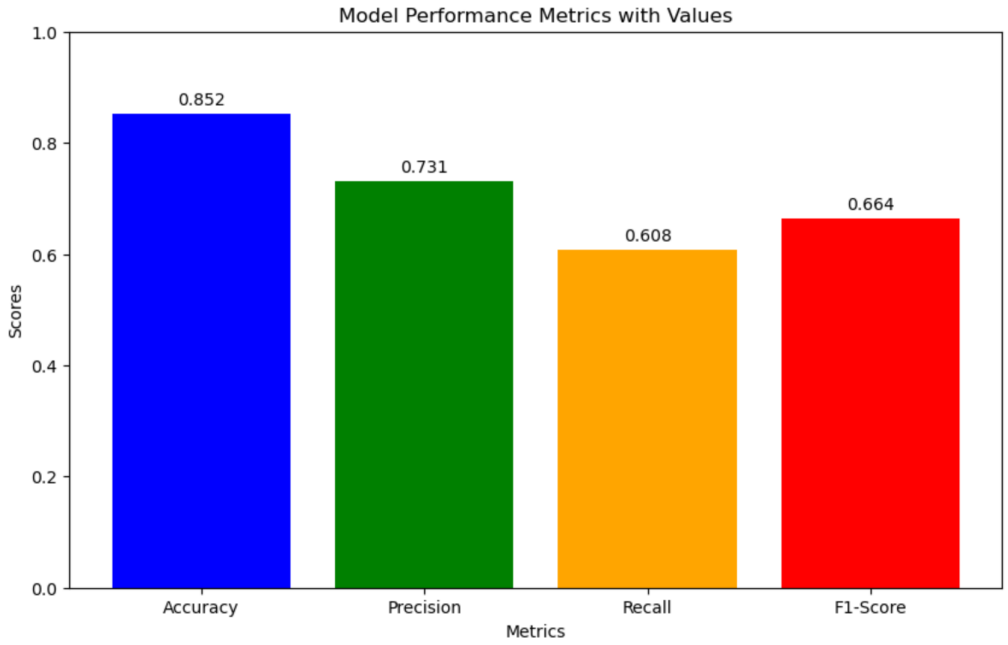


Fig 13

From the graph of the Random Forest classification model:

The model correctly predicts the outcome 85.2% of the time, indicating a relatively high level of overall performance. When the model predicts an instance as positive, it is correct approximately 73.1% of the time. This metric suggests that the model has a moderate number of false positives. When it comes to recall, the model correctly identifies 60.8% of all actual positive cases. This is the lowest score among the three metrics, indicating that a significant number of positive cases are being missed (false negatives). F1-Score (0.664): The F1-score, which combines precision and recall, is moderate, indicating that the model has a fair balance between precision and recall. However, this score also reflects that there is room for improvement, especially in correctly identifying positive cases.
In summary, the Random Forest model performs well in terms of accuracy, but there is a noticeable drop when it comes to recall and F1-score, which could be due to class imbalance or other factors that affect the model's ability to detect positive cases.

Tunning the parameters:
Now we use the GridSearchCV package in Python to check which parameter fits the best Random Forest model. The param grid we set is shown below. We will also do a 5-fold CV in the Grid Search.

```
        max_depth min_samples_split min_samples_leaf      bootstrap
0  [4, 6, 8, 10]        [2, 5, 10]        [1, 2, 4]  [True, False]
```

Fig 14

```
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best parameters found:  {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2}
Best score found:  0.8588794363382423
```

Fig 15

The grid search tested each combination of parameters (72 in total) with 5-fold cross-validation. The best-performing combination of parameters includes not using bootstrap samples ('bootstrap': False), setting the maximum depth of the trees to 10 ('max_depth': 10), requiring at least 4 samples to form a leaf node ('min_samples_leaf': 4), and requiring at least 2 samples to split a node ('min_samples_split': 2). The score associated with the best-performing model is approximately 0.858, which is slightly greater than the model without tunning.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a statistical technique used for classification and dimensionality reduction. It seeks to find a linear combination of features that best separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or for dimensionality reduction before later classification.

Data Preparation:

1. Removing NaN value like described above.

2. *Dummy variable adjustment:*

The dataset included both numerical and categorical variables, was preprocessed. Categorical variables were all encoded using dummy-variable encoding, converting them into a binary matrix necessary for the model.

workclass_Local-gov	workclass_Never-worked	workclass_Private
False	False	True
False	False	True
False	False	True
False	False	True
False	False	True

Fig 16

The labelled dependent variables are also divided into two groups with encoding, with income "<= 50k" being "0" and "> 50K" being "1"

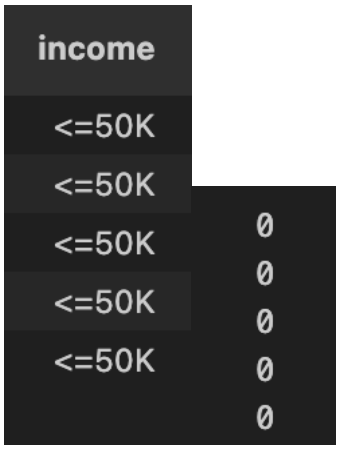


Fig 17

Model Training and Validation:
The model was then instantiated and evaluated using 5-fold cross-validation to estimate its accuracy. This method partitions the data into 5 sets, iteratively using one set for validation and the remaining for training, thus providing a robust measure of the model's predictive power.

Performance Evaluation:
The model's performance was quantified in terms of cross-validation accuracy, which reflects the proportion of correct predictions made by the model over the entire dataset.

```
Confusion matrix:
[[22810  1910]
 [ 3649  4192]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.92      0.89     24720
           1       0.69      0.53      0.60      7841

    accuracy                           0.83     32561
   macro avg       0.77      0.73      0.75     32561
weighted avg       0.82      0.83      0.82     32561
```

Fig 18

Fig 19

Confusion Matrix:

- It correctly predicted 22,810 individuals as earning "<=50K" (true negatives).
- It incorrectly predicted 3,649 individuals as earning "<=50K" (false negatives).
- It correctly predicted 4,192 individuals as earning ">50K" (true positives).
- It incorrectly predicted 1,910 individuals as earning ">50K" (false positives).

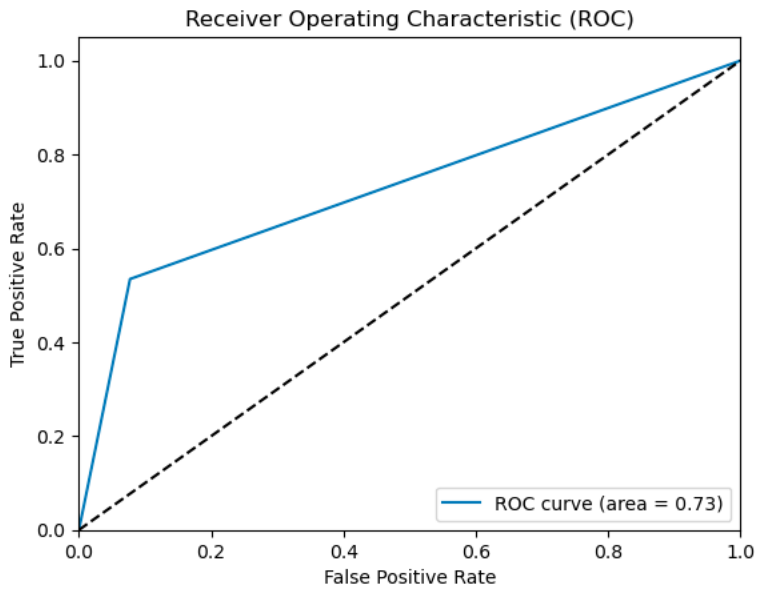
Classification Report:
For individuals earning "<=50K":
Precision: 86% of individuals predicted to earn "<=50K" actually do.
Recall: 92% of the actual individuals earning "<=50K" were correctly identified.
F1-Score: A high score of 89%, indicating a good balance between precision and recall for this income bracket.

For individuals earning ">50K":
Precision: 69% of individuals predicted to earn ">50K" actually do.
Recall: 53% of the actual individuals earning ">50K" were correctly identified.
F1-Score: A moderate score of 60%, showing that the model is less effective at identifying individuals in this higher income bracket compared to the "<=50K" bracket.
Accuracy: The overall accuracy of the model is 83%, which suggests it is relatively effective at classifying individuals across the two income levels.

Macro Average:
Precision: 77% average precision across both income levels.

Recall: 73% average recall, indicating the model more accurately identifies individuals earning "<=50K".
F1-Score: 75% average F1-score, demonstrating a balanced performance with a lean towards the "<=50K" income group.

Weighted Average:
Precision: 82% weighted precision, considering the distribution of the income groups.
Recall: 83% weighted recall, also adjusted for the prevalence of each income group.
F1-Score: 82% weighted F1-score, indicating overall balanced performance when accounting for class imbalance.



The AUC score of 0.73 indicates that the classifier has a good ability to distinguish between the positive class (likely ">50K" income) and the negative class (likely "<=50K" income). In general, an AUC score above 0.7 is considered acceptable, but there is room for improvement.

Fig 20

The LDA model has a stronger performance in identifying individuals earning "<=50K" compared to those earning ">50K," as reflected by the higher metrics for the "<=50K" group. This suggests a potential area of focus for model improvement could be increasing the correct identification of individuals earning ">50K".

Conclusion:

After evaluating various models, the Random Forest stands out with the highest accuracy, indicating a strong ability to correctly classify individuals across income brackets. Despite the solid performance of the Linear Discriminant Analysis in predicting lower-income earners, it underperforms in identifying higher-income individuals. The Generalized Additive Model and Logistic Regression, while considered, did not outperform the Random Forest in overall accuracy. Therefore, the Random Forest model is selected as the preferred model due to its superior accuracy and robustness across different performance metrics.

Code Repo:

Released in: https://github.com/pengyuyang-315/MDS583_CODE.git