Classifying Income Levels Using Census Data

By: Hanumantha Sai Karthik Velavarthypathi

Abstract

This project explores the application of classification algorithms to predict income levels based on data from the U.S. Census Bureau. Using the Census Income dataset, the objective is to classify individuals into two income levels: those earning over $50K annually and those earning $50K or less.

This project involves preprocessing the dataset to handle missing values, encode categorical variables, and address class imbalance. Multiple classification algorithms, including KNN, Naive Bayes, Random Forest, and an ensemble model having all of these three models, are implemented and evaluated based on accuracy, precision, recall, and F1-score metrics.

Dataset Preprocessing

Dataset Description

The dataset used for this project is the **Adult Income Dataset**, obtained from the UCI Machine Learning Repository. The dataset is widely used for classification tasks, precisely predicting whether an individual earns more than $50K/year based on demographic and employment-related attributes.

Features:

1. **Categorical Attributes:**
    - **Workclass**: Type of employment (private, self-emp-not-inc, self-emp-inc, federal-gov, local-gov, state-gov, without pay, never worked).
    - **Education**: The highest level of education achieved (Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool).
    - **marital_status**: Marital status (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse).
    - **Occupation**: Occupation type (Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspect, Adm- clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces).
    - **Relationship**: Relationship to household (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried).
    - **Race**: The individual's race (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black).
    - **Sex**: Gender (Male, Female).
    - **Native_country**: Country of origin (United States, Cambodia, England, Puerto Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary,

Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands).
    - **Income**: Target variable, indicating <=50K or >50K.
2. **Numerical Attributes:**
    - **Age**: Age of the individual.
    - **Fnlwgt**: Final weight represents the number of people to whom the census information applies.
    - **Education_num**: Numerical representation of education level.
    - **Capital_gain**: Income from investment sources other than wages.
    - **Capital loss**: Losses from investments.
    - **Hours_per_week**: Weekly working hours.

```
Adult Data (Train):
   age         workclass  fnlwgt  education  education-num  \
0   39         State-gov   77516  Bachelors             13
1   50  Self-emp-not-inc   83311  Bachelors             13
2   38           Private  215646    HS-grad              9
3   53           Private  234721       11th              7
4   28           Private  338409  Bachelors             13

       marital-status         occupation   relationship   race     sex  \
0       Never-married       Adm-clerical  Not-in-family  White    Male
1  Married-civ-spouse    Exec-managerial        Husband  White    Male
2            Divorced  Handlers-cleaners  Not-in-family  White    Male
3  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
4  Married-civ-spouse     Prof-specialty           Wife  Black  Female

   capital-gain  capital-loss  hours-per-week native-country income
0          2174             0              40  United-States  <=50K
1             0             0              13  United-States  <=50K
2             0             0              40  United-States  <=50K
3             0             0              40  United-States  <=50K
4             0             0              40           Cuba  <=50K
```

Figure: screenshot of data sample before any preprocessing

Preprocessing Steps

1. Handling Missing Values

- **Categorical Columns:** Missing values were replaced with each column's most frequent category (mode).
- **Numerical Columns:** Missing values were filled with the median of the respective columns.

2. Encoding Categorical Variables

- Converted all categorical variables into a numeric format using **Label Encoding**, which assigns a unique integer to each category.

3. Balancing the Dataset

Figure: Class Distribution among the two classes in the training data

- Addressed class imbalance in the target variable (income) using **SMOTE (Synthetic Minority Oversampling Technique)** to generate synthetic samples for the minority class. This prevents the model from being biased towards one particular class.

income	count
0	24720
1	24720

Figure: The number of samples in each class after applying SMOTE

4. Target Variable Encoding

- Converted the target variable (income) into binary format: <=50K: 0 and >50K: 1

Methodology

Naive Bayes model

The naive Bayes model is a probabilistic classifier that uses the Bayes theorem to calculate the probability of the hypothesis of the given data. It assumes that the features are all conditionally independent of each other.

$$Y^{\text{predict}} = \underset{v}{\text{argmax}}\, P(Y = v) \prod_{j=1}^{n_Y} P(X_j = u_j \mid Y = v)$$

The Gaussian Naive Bayes model was chosen for its simplicity and effectiveness on moderately sized datasets. The model was initialized using the training dataset ('X_train' and 'y_train') to predict income categories.

KNN model

K-nearest neighbors (KNN) algorithm stores training sample and calculates the distances to find k neighbors to the new data point, and based on that, it predicts the class of the latest data point. KNN is a simple yet effective instance-based learning algorithm.

The K-Nearest Neighbors (KNN) algorithm was used to classify income levels based on the dataset. I used Euclidean to calculate distance. The default value of k=5 was used. It predicts the class of a sample based on the majority class of its five nearest neighbors in the feature space.

$$\text{Euclidean Distance} \qquad \sqrt{(\sum_{i=1}^{k}(x_i - y_i)^2)}$$

Random Forest model

The Random Forest Algorithm model is an ensemble of many decision trees where the random sampling of data is constructed into decision trees, and based on the majority class of those decision trees is the classification of the random forest model.

$$\mathbf{f(x) = (1/T) * \Sigma(t=1 \text{ to } T) [dt(x, \Theta t)]}$$

Where:

- **f(x)**: is the final prediction for input data point "x."
- **T**: is the total number of decision trees in the forest.

- **Dt (x, Θt)**: represents the prediction made by the t-th decision tree on input "x" using its randomly selected features (Θt).

The Random Forest Classifier using the RandomForestClassifier class from the sklearn library. The parameter n_estimators=100 specifies the number of decision trees in the forest, ensuring a robust model by averaging the outputs of 100 trees. The random_state=42 parameter sets a seed for reproducibility, allowing the same results to be obtained across different code runs.

Ensembled model

The ensemble model combines multiple machine-learning models to make a final class prediction. Its ability to mix various machine learning models improves the model's overall performance as it takes the best of the models it has used.

$$\text{Final Prediction} = (\Sigma(\alpha\_i * h\_i(x)) / N)"$$

Where:

- "α_i" represents the weight assigned to each individual model "h_i" within the ensemble.
- "h_i(x)": does the i-th model predict input data "x."
- "N" is the total number of models in the ensemble.

This ensemble approach leverages the strengths of different models to improve classification accuracy and generalizability. Soft voting ensures that probabilistic predictions are considered, making the model more robust for this binary classification problem.

The project has implemented the K-Nearest-neighbours, Naive Bayes, and Random Forest in the ensemble model, so the majority class out of these models is the predicted class of the ensemble model.

The VotingClassifier combines three distinct models—Naive Bayes, K-Nearest Neighbors (KNN), and Random Forest—into a unified ensemble to classify income levels. Soft voting averages the probabilistic outputs of these models to make the final prediction, leveraging their complementary strengths for improved accuracy and robustness. The model's performance was evaluated using accuracy, precision, recall, F1 score, specificity, and ROC AUC. Visualization tools, including a confusion matrix and an ROC curve, provided more profound insights into the classifier's effectiveness. By integrating diverse algorithms, the ensemble capitalized on probabilistic and distance-based learning and decision-tree ensembles to enhance classification performance.

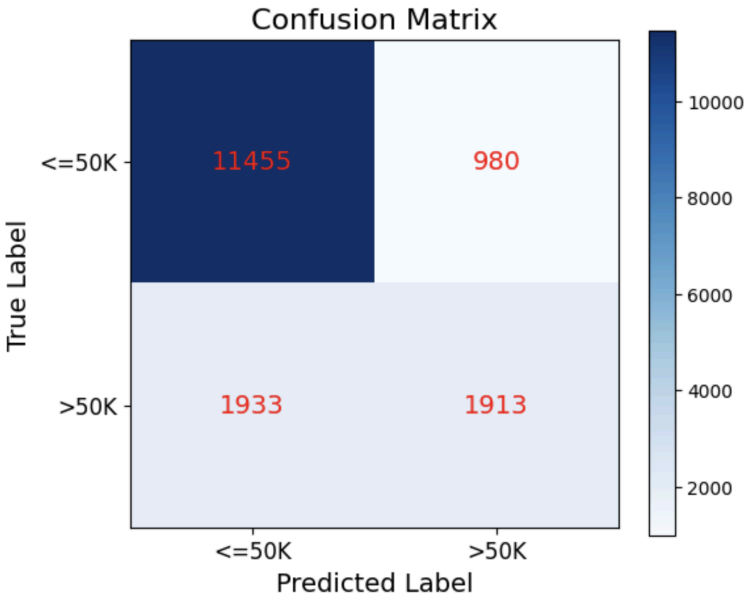
Results and analysis

Results of the naive Bayes model:

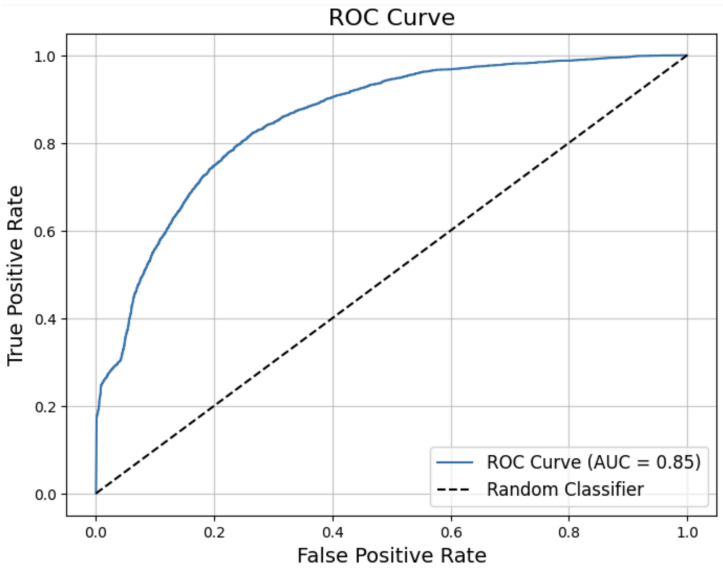
The naive Bayes model has been good overall, as indicated by its accuracy. This model is also good at predicting the less than or equal to 50k class, as noted in the confusion matrix, but it is not the best in predicting the other class as well as indicated by recall and roc score.

Training Accuracy: 0.7116100323624596
Testing Accuracy: 0.8210797862539156
Precision: 0.6612512962322848
Recall: 0.49739989599583984
F1 Score: 0.5677400207745956
Specificity: 0.9211901889827101

Confusion Matix:



Roc-Auc curve:

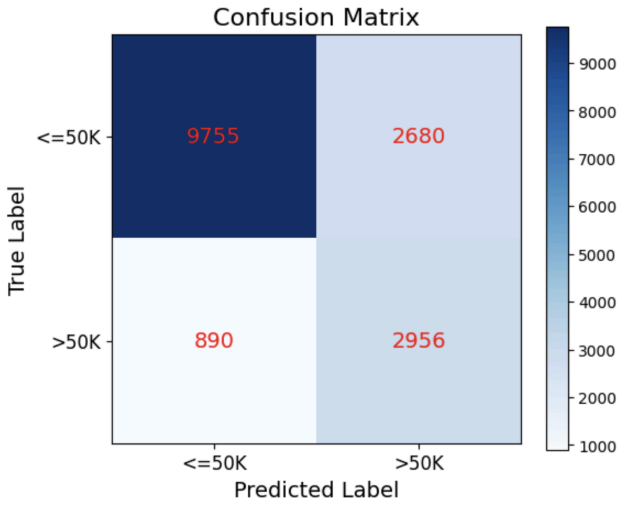


Results of the KNN model:

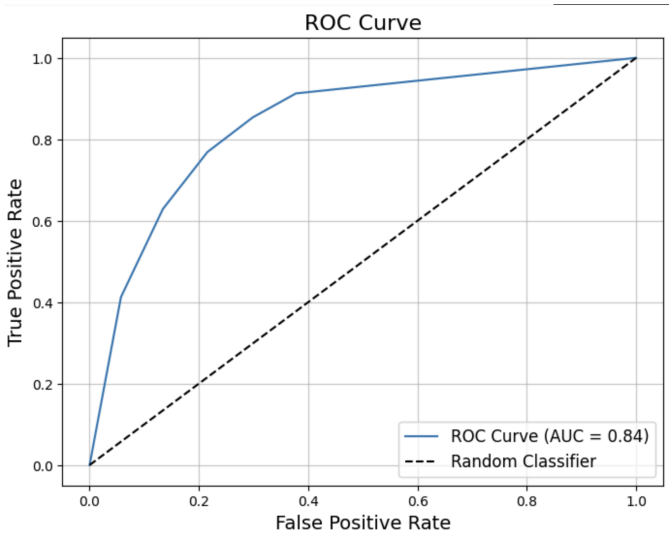
The KNN model has a good overall performance, as indicated by accuracy, but it is still the weakest of the three models. It has a higher recall compared to the naive Bayes model. This model doesn't have good enough precision compared to other models.

Training Accuracy: 0.905542071197411
Testing Accuracy: 0.7807259996314723
Precision: 0.524485450674237
Recall: 0.7685907436297452
F1 Score: 0.6234971524994727
Specificity: 0.7844792923200643

Confusion matrix:



Roc-AUC curve:

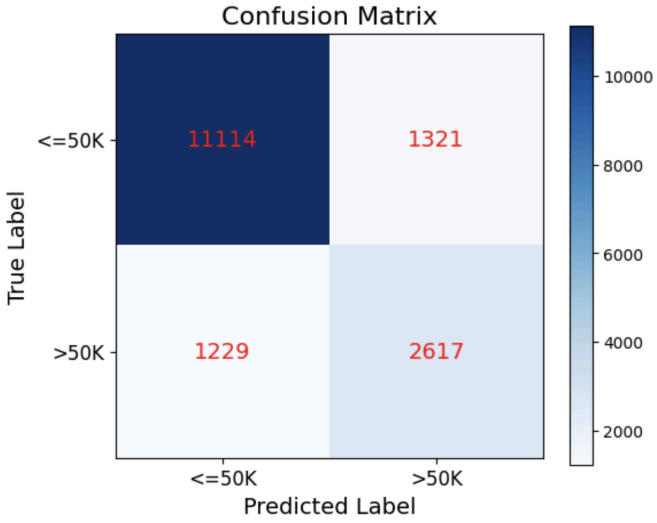


Results of Random Forest:

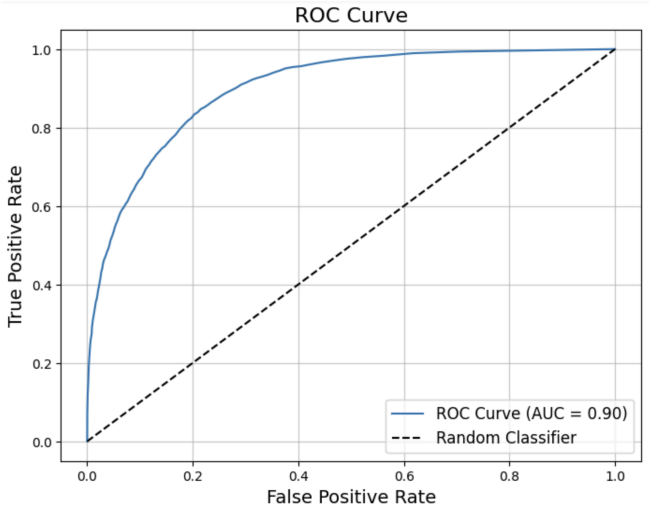
The Random Forest model has the best accuracy of the three models and has the most substantial results compared to the rest of the models.

Training Accuracy: 0.9999190938511326
Testing Accuracy: 0.8433757140224802
Precision: 0.6645505332656171
Recall: 0.6804472178887155
F1 Score: 0.6724049331963001
Specificity: 0.8937675914756735

Confusion Matrix:



Roc AUC curve:



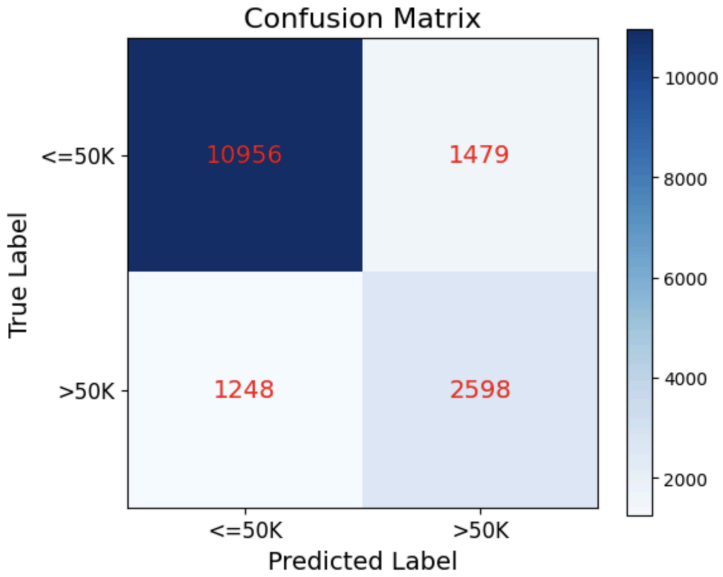
Results of Ensemble model:

The Ensemble model has improved the overall performance metrics compared to the individual models, except for the Random Forest. The performance metrics are more robust than those of the

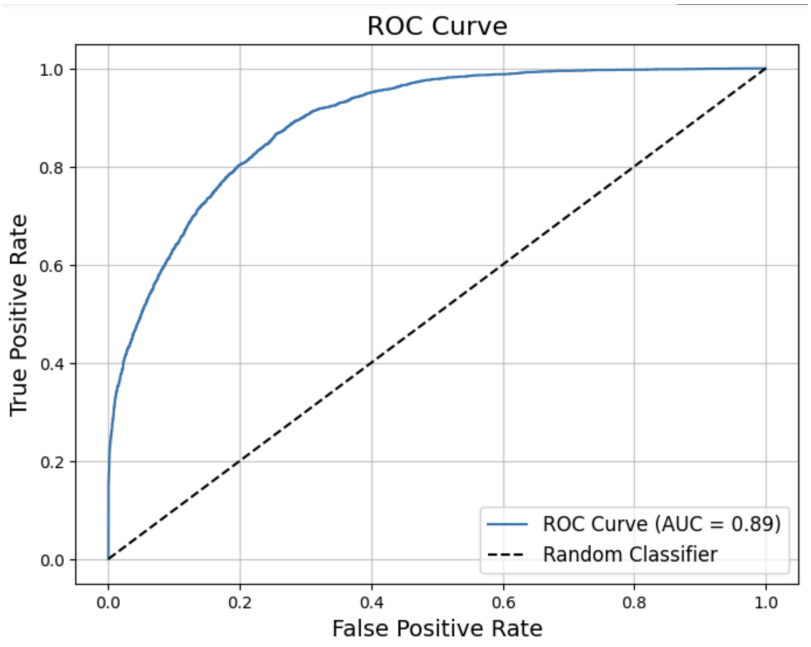
KNN and Naive Bayes individual models, but the ensembled model is weaker than the Random Forest model. The ensemble model is strong and has high testing accuracy. This model also has better precession and recall than the KNN and Naive Bayes individual models and is good at predicting the income class.

Training Accuracy: 0.9592435275080906
Testing Accuracy: 0.8325041459369817
Precision: 0.637233259749816
Recall: 0.6755070202808112
F1 Score: 0.6558121923513821
Specificity: 0.8810615199034982

Confusion matrix:



Roc AUC curve:



Conclusion

The missing values in the dataset have been handled by using mode for categorical values and mean for continuous or numerical values. Label Encoding has been used to convert the categorical values into numerical values. Label encoding was done to keep the numerical value constant across the same category. To address the imbalance in the training data between the classes of more than fifty thousand and less than or equal to fifty thousand dollars of annual income, SMOTE (Synthetic Minority Oversampling technique) is deployed to create artificial samples of data to make the balance between the two classes. The ability to handle missing data, use label encoding, and apply SMOTE has improved how the data can be used and helps avoid losing any data. SMOTE helped make the dataset more balanced to help the model learn better.

The random forest model has the best evaluation metrics from the KNN, Naive Bayes, random forest, and ensembled models. The Random Forest performs better than the ensembled model at most metrics. The Ensemble enhanced the model's overall performance compared to individual models like naive Bayes and knn.

References

- Google Colab
- UCI Machine Learning Repository
- Naive Bayes
- KNN
- Random Forest
- Ensemble model
- SMOTE