Prediction of Annual Income

A Comparative Study of Logistic Regression, SVM, and Gradient Boosting Classifier

Wang Wenxin, G2304475D

IN6227-2023-Assignment-1.2

INTRODUCTION

The goal of this assignment is to make a prediction based on 1994 US Census to predict whether a person's annual income will exceed $50,000.

In this assignment, three different models have been employed for comparison: Logistic Regression (LR), Support Vector Machine (SVM), and Gradient Boosting Classifier (GBDT). This report will also discuss the data pre-processing steps, hyperparameter tuning, model evaluation, and provide insights into the performance of these models.

DATA PREPROCESSING

DATA LOADING

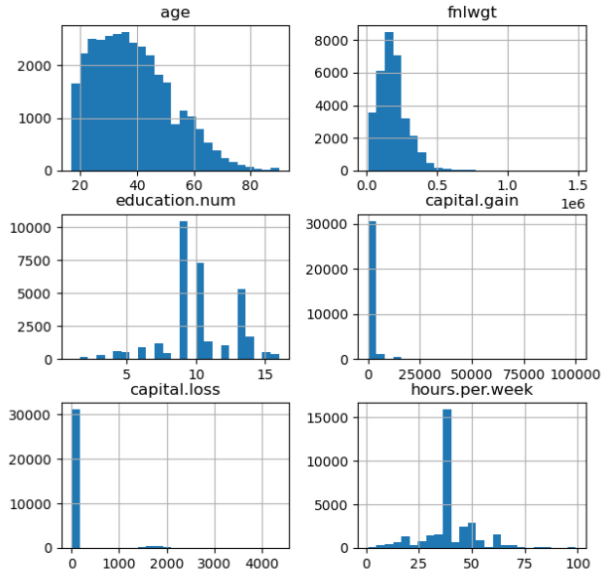
The code starts by loading the dataset from files named 'adult.train' and 'adult.test' using Pandas. Two DataFrames are created: 'df_train' and 'df_test', which contain data from training and testing sets, respectively.

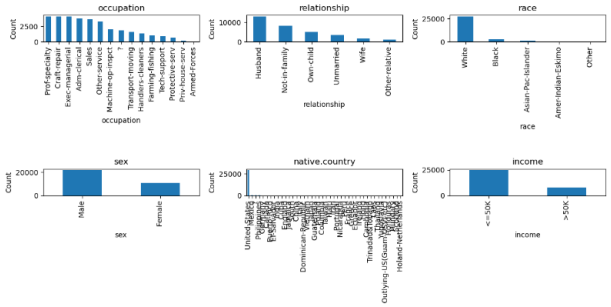
HANDLING MISSING VALUES

The code handles missing values ' ?' by replacing them with 'NaN' using the replace method. Afterward, rows with missing values are dropped from the datasets using the '.dropna()' method.

FEATURE ENGINEERING

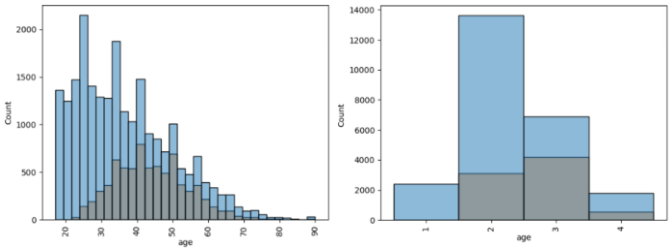
To enhance the analysis of raw data, infographics for numerical features and categorical features is applied. It can be seen that there is a significant imbalance in the income distribution. Therefore, employing oversampling or undersampling techniques to rebalance the income data may be beneficial to the training outcome.





It should be noticed that some of the features may have too many values and can make negative impacts to the training. Therefore, we can create classification features, which may be more useful in some ML models. Due to space constraints, only the results of re-classification of the feature 'age' are shown here.

- Age Grouping: The age_group function is defined to group ages into four categories based on predefined bins. This grouped feature is one-hot encoded and added to the datasets.



- Workclass: The 'workclass' feature is transformed into a binary feature, where 'Private' is encoded as 1, and other values as 0.
- Education: The 'education' feature is one-hot encoded, and 'education.num' is dropped as it provides similar information.
- Categorical Features: All other categorical features ('marital.status', 'occupation', 'relationship', 'race', 'sex', 'native.country') are one-hot encoded to convert them into a numeric format.
- Scaling: Numerical features ('capital.gain', 'capital.loss', 'hours.per.week') are standardized using StandardScaler.

MODEL BUILDING

Three models are built to finish the prediction task:

- Logistic Regression (LR)
- Support Vector Machine (SVM)
- Gradient Boosting Classifier (GBDT)

MODEL EVALUATION

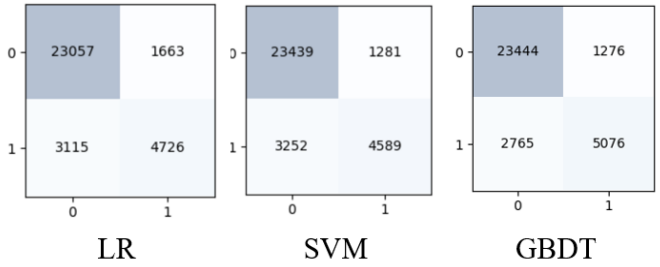
The code performs cross-validation on the training data for each model and prints the mean accuracy and

standard deviation. The model with the highest mean accuracy is selected as the best model. In this case, the **Gradient Boosting Classifier** appears to be the best model. The comparison of the accuracy and training time of the 3 models are summarized in the table below:

MODEL	Acc	Training Time(s)
LR	0.853	0.67
SVM	0.861	48.94
GBDT	0.876	16.49

- Confusion Matrix

A confusion matrix is visualized to understand the model's performance on the test data.



- Model Metrics

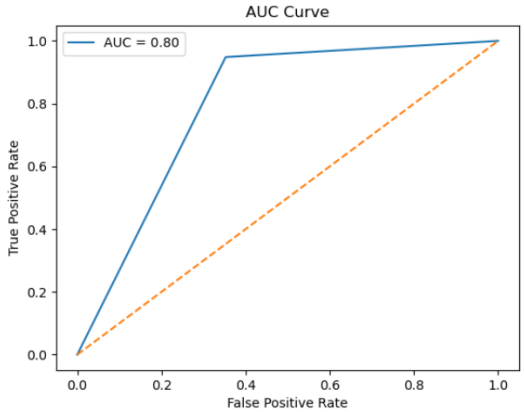
The code computes various performance metrics for the best model, encompassing critical measures such as Accuracy (overall correctness), Precision (positive predictions' accuracy), Recall (true positive capture rate), and F1-Score (harmonic mean of Precision and Recall).

MODEL	LR	SVM	**GBDT**
Accuracy	0.853	0.861	0.876
P	0.881	0.878	0.895
Recall	0.933	0.948	0.948
F1	0.906	0.912	0.921

- AUC Curve

Curve, or Area Under the Curve, is a graphical representation of a model's ability to distinguish between positive and negative classes in a binary classification problem. It measures the model's overall performance, with a higher AUC indicating better discrimination.

The AUC curve for the best-performing GBDT model among the three models is as follows.



DISCUSSION

TRAINING TIME

Training time is an important consideration in model selection and deployment. The Support Vector Machine (SVM) had the longest training time, taking almost 49 seconds. This prolonged training period might not be ideal in real-time applications or large-scale datasets. On the other hand, Logistic Regression and the Gradient Boosting Classifier had significantly shorter training times, with the latter taking approximately 16.49 seconds. Therefore, the Gradient Boosting Classifier not only performed better but also trained relatively faster compared to SVM.

MODEL SELECTION

Choosing the right model depends on a trade-off between accuracy and training time. If time is not a critical factor and the highest accuracy is desired, the Gradient Boosting Classifier is the best choice. However, if faster training and a slightly lower accuracy are acceptable, Logistic Regression might be a more practical option. SVM, despite its longer training time, offers competitive accuracy, making it a viable choice if computational resources are available.

FUTURE IMPROVEMENTS

To further enhance model performance, several steps can be taken:

- Hyperparameter Tuning: Fine-tune hyperparameters of the Gradient Boosting Classifier and SVM to optimize their performance.
- Feature Engineering: Explore more advanced feature engineering techniques to potentially improve model predictions.
- Data Expansion: Collect additional data or use data augmentation techniques to increase the size of the dataset.

CONCLUSION

In conclusion, the code provided loads the data, preprocesses it, and trains three different models to predict annual income. The Gradient Boosting Classifier outperforms the other models in terms of accuracy. The model evaluation metrics provide valuable insights into the model's performance. Further tuning and optimization could potentially improve the model's accuracy, but the provided code serves as a solid starting point for income prediction based on the given dataset.

CODE RESOURCE

Link: jiushuw/IN6227-Assignment1.2 (github.com)

REFERENCES

[1] Friedman, J.H. (2001) 'Greedy function approximation: A gradient boosting machine.', The Annals of Statistics, 29(5). doi:10.1214/aos/1013203451.

[2] Platt, J.C. (1998) 'Fast training of support vector machines using sequential minimal optimization', Advances in Kernel Methods [Preprint]. doi:10.7551/mitpress/1130.003.0016.