CS311 Project3 Report: Effectiveness Evaluation on Machine Learning Models for solving Census Prediction Problem

Zhengdong Huang *12212230*

***Abstract*—This article is a project report of CS311 Project 3. In this report, I evaluate the performance of machine learning models in solving Census Prediction Problem. Also, I conducted experiment in evaluating the effectiveness of various data preprocessing approaches. The results demonstrate that the XGBoost model perform best in census prediction with high accuracy and f1-score, which is finally selected as the prediction model.**

***Index Terms*—Machine Learning Algorithm, Census Prediction**

I. Introduction

THE Adult Census Income Prediction Problem is a well-known task in the field of machine learning and data science. As a typical classification problem, this problem has its roots in the UCI Machine Learning Repository [1] and has been widely used as a benchmark dataset for evaluating classification algorithms.

The background of this problem dates back to the early 1990s when the dataset was extracted from the 1994 Census database by Barry Becker containing about 30 thousands of entries [1]. The dataset includes various attributes representing the basic information of each individuals, which are used to predict whether the individual's annual income exceeds $50,000 [2].

Applications of this problem are vast and varied. It is primarily used in academic research and education to teach and evaluate machine learning models. Additionally, due to to the high social relevance of the dataset, it has practical applications in areas such as social science research, economic studies, and policy-making. By analyzing the factors that contribute to higher income, researchers and policymakers can develop strategies to address income inequality and improve economic conditions [6].

Current research in this area focuses on developing and comparing various machine learning algorithms to achieve higher accuracy in solving classification problem. Classical Classification techniques such as logistic regression, decision trees, random forests, support vector machines have been extensively studied. Additionally, researchers are exploring advanced methods like ensemble learning, deep learning to enhance the predictive power of models. There is also a growing interest in addressing fairness and bias in predictions, trying to develop a more applicable model [12]. As a typical classification problem, the above techniques might possible in solving Revenue Prediction Problem. [4]

This paper will primarily evaluate the effectiveness on various machine learning model in solving Revenue Prediction Problem. It will delve into the exploration on preprocessing techniques, and evaluation on 9 classical machine learning model in solving the problem, and summarize the possible best approaches for problem solving. We seek to elucidate the effectiveness of our methodology for problem solving workflow and final solution.

II. Preliminary

The Revenue Prediction Problem can be formulate as follows:

Given an individual info X, with multiple attributes $X = \{x_1, x_2, ...x_n\}$ representing age, education and so on, the problem is to find a mapping $f : X \to Y$, where $Y = \{1, 0\}$, which represent whether the annual income of individual is larger than $50 K or not. Based on the provided dataset, the problem further aims to maximize the accuracy of mapping, i.e. correctly predicting the income class of each individual.

III. Methodology

Our research is centered around the exploration of Data Preprocessing Techniques, Model Selection and Final Evaluation. The workflow of this method will be presented in subsection A, followed by a detailed explanation of the fundamental steps of in subsection B. The explanation of model selection will be presented in subsection C, with the Optimality analysis will be further detailed in subsection D.

A. General Workflow

To find the best preprocessing techniques and models that can maximize the accuracy, we adopted the following workflow for exploration, as shown in fig.1.

B. Data Preprocessing

1) Data Analyzing: **Distribution Analysis:** Base on the given dataset, it is important to analyze the characteristics of it, which can effectively guide the preprocessing of the datasets. The provided training dataset has 22792 entries, with 14 properties for each entry. The characteristic of the properties is shown as fig.2. We further analyze and visualize the distribution of the dataset respect to properties, where we discovered important characteristics that can guide the following preprocessing steps.

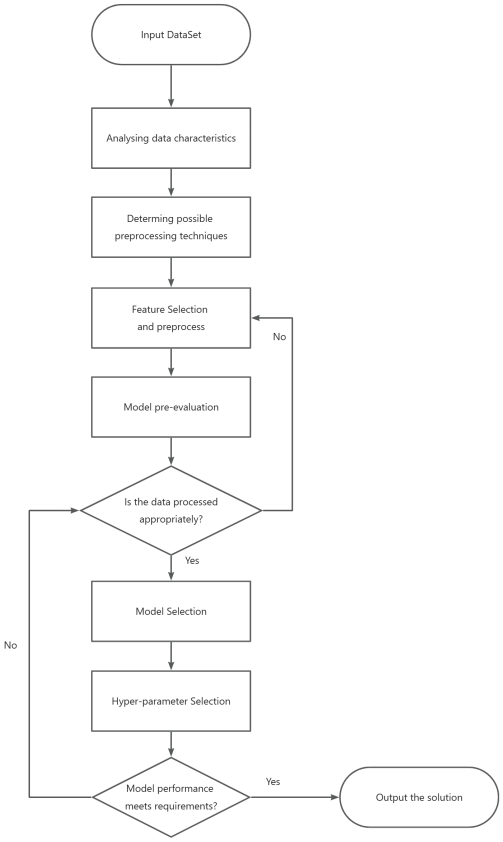
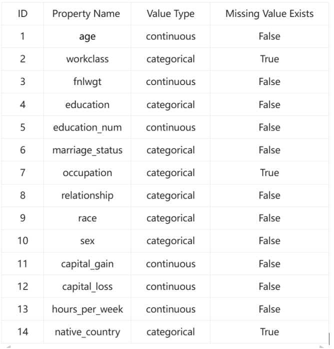

Input DataSet
Analysing data characteristics
Determing possible preprocessing techniques
Feature Selection and preprocess
No
Model pre-evaluation
Is the data processed appropriately?
Yes
Model Selection
No
Hyper-parameter Selection
Model performance meets requirements?
Yes
Output the solution

Fig. 1: General Workflow of Research



ID	Property Name	Value Type	Missing Value Exists
1	age	continuous	False
2	workclass	categorical	True
3	fnlwgt	continuous	False
4	education	categorical	False
5	education_num	continuous	False
6	marriage_status	categorical	False
7	occupation	categorical	True
8	relationship	categorical	False
9	race	categorical	False
10	sex	categorical	False
11	capital_gain	continuous	False
12	capital_loss	continuous	False
13	hours_per_week	continuous	False
14	native_country	categorical	True

Fig. 2: Properties characteristics of each entry

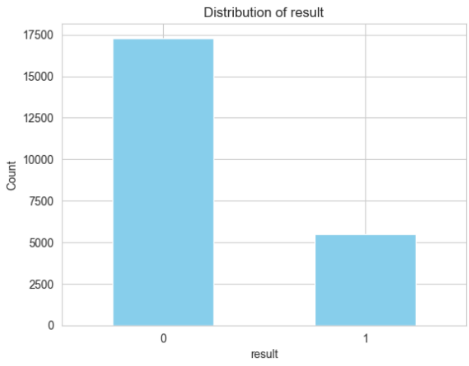


Fig. 3: Result Distribution

As illustrated in fig.3, this is an unbalanced dataset that the ratio of result type 0 to 1 is about 76:24. For unbalanced datasets, in addition to focusing on the accuracy of the overall model, it is also necessary to focus on the performance of the predictions in the less trained part (i.e., with a result of 1). As discussed in following sectionsIV, we will adopts various evaluation criteria to analyze the model performance. Also, we set a baseline of the prediction accuracy as 76%, where prediction accuracy higher than it is considered as a valuable predicting model.

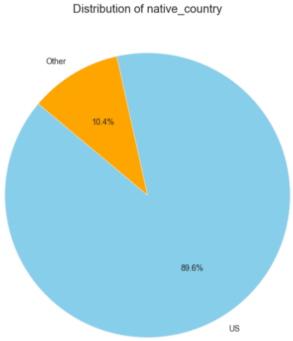
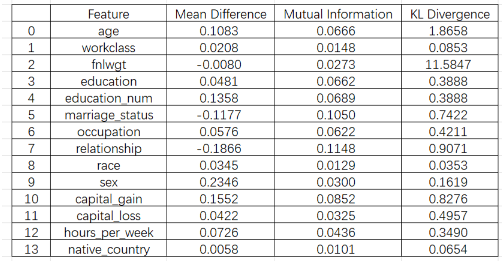


Fig. 4: Native Country Distribution

Also, the imbalance is also observed for each property. Some of the properties with minor imbalance is acceptable. However, we also observed a extreme unbalance in properties like ”native_country”. As shown in fig.4, over 90% of the ”native_country” of individuals in United-States, while the remaining 10% are from about 40 different nationalities. As will be mentioned below, too much variety can make coding difficult, especially one hot coding, so we propose a possible way of data preprocessing:

1. Classification of nationality as US and other

Relative Analysis: To better process the dataset, it is important to analyze the relationship between properties and results. There are a number of different metrics used in correlation analysis, and we mainly used Mean Differences, Mutual Information and Kullback-Leibler Divergence (KL Divergence) for analysis between attributes and results:



	Feature	Mean Difference	Mutual Information	KL Divergence
0	age	0.1083	0.0666	1.8658
1	workclass	0.0208	0.0148	0.0853
2	fnlwgt	-0.0080	0.0273	11.5847
3	education	0.0481	0.0662	0.3888
4	education_num	0.1358	0.0689	0.3888
5	marriage_status	-0.1177	0.1050	0.7422
6	occupation	0.0576	0.0622	0.4211
7	relationship	-0.1866	0.1148	0.9071
8	race	0.0345	0.0129	0.0353
9	sex	0.2346	0.0300	0.1619
10	capital_gain	0.1552	0.0852	0.8276
11	capital_loss	0.0422	0.0325	0.4957
12	hours_per_week	0.0726	0.0436	0.3490
13	native_country	0.0058	0.0101	0.0654

Fig. 5: Relationship between Properties and results

As illustrated above, there is some differences in the degree of association between different attributes and outcomes. Attributes like education_num, age and relationship are higher relative with result than other attributes. It is important to notice the extreme high KL divergence in attribute fnlwgt,

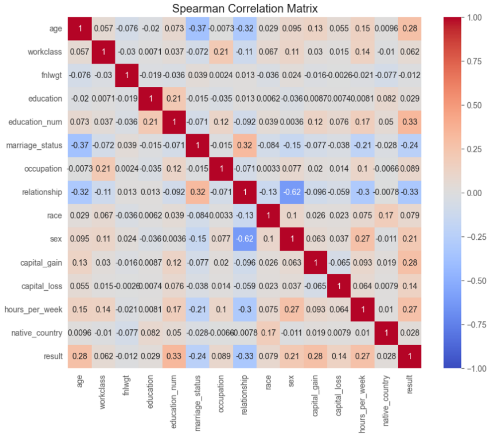


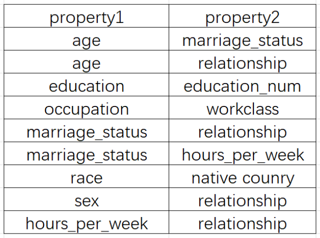
Fig. 6: Spearman Correlation Matrix

which, however, has a low value on MD and MI. This might due to the unbalance of dataset and the wide value range of fnlwgt. The actual meaning to fnlwgt is ambiguous. In some studies, this value is considered to be the weighted sum result of the attributes of the state statistical offices, and fnlwgt that are similar in the same state will have each of their attributes will be similar [14]. Considering it, we proposed another possible way of data preprocessing:

2. Remove column fnlwgt.

In addition to the associations between each attribute and the results, the degree of association between each attribute that we further analysed. We selected the spearman correlation matrix as indicators of relationship. Spearman Correlation Matrix is a matrix that displays the Spearman rank correlation coefficients between multiple variables. It has lower time complexity and can evaluate the relationship under unbalanced dataset that not follow the normal distribution [13].

As illustrated in fig.7, the following property pairs has high relationship between each other.



property1	property2
age	marriage_status
age	relationship
education	education_num
occupation	workclass
marriage_status	relationship
marriage_status	hours_per_week
race	native counry
sex	relationship
hours_per_week	relationship

Fig. 7: High Relative Pairs

Strong correlations between columns such as age, relationships, marriage.status, etc. are in line with our basic knowledge. Further checking demonstrate the possible combination of those pairs are large, resulting the difficulty in merging columns. However, for pair education,education.num, there are only several possible combinations. Since there is only one to one mapping between education and education.num, it is necessary for removing one attributes of them to reduce the duplication affect the model. The education.num is continuous without requirement for encoding, which is better to reserve. Also, we observed that there is no individuals who has both positive capital gain and capital loss, which satisfied our life knowledge. We can merge the two column to reduce the duplication in columns. Thus,we proposed possible way of data preprocessing:

3. Remove column education.

4. For each individual, subtract their capital gain with capital loss, and remove capital loss column.

2) Preprocessing: The general data preprocessing workflows is described as follows: 1. Remove duplicate columns in dataset. For machine learning algorithm, it is important to reduce the duplicate columns, which can increase the independence between attributes [10]. For this problem, we mainly adopt steps proposed in the data Analyzing section to remove duplicate columns.

2. Handling Missing Value.

As illustrated in above section, various properties contains unknown values, which is unable to be handled by distance based Model like SVM [7]. There are various ways in preprocessing the missing value. Thus, we decided to fill the missing value as the mode in the dataset.

3. Encoded Discrete Properties:

Since most of the values in discrete properties are string, which cannot be handled by most of the machine learning algorithm, it is important to convert the value into number. Labelling strings directly may introduce unnecessary sequencing, which can have implications for the performance of many machine learning models.so we used One hot encoding for the discrete string variables [11].

4. Value Processing: Value processing includes approaches such as Normalisation the attributes, which can reduce the impact of the unit scale, enhancing distance-based machine learning models. Thus, we tried the Normalization in preprocessing stages.

5. Dimensional Reduction: For datasets with a large number of dimensions, dimensional reduction is an important operation. This not only significantly reduces the computational cost, but also better enables the columns to be independent of each other and highlights valuable information, thus improving the overall machine learning model. In our research, we attempted Principal Component Analysis(PCA) algorithm [9], which can effectively reduce the dimension in large datasets.

The above preprocessing operations can be combined to produce a large number of feasible preprocessing schemes, and in our study, we manually selected a set of better preprocessing schemes with 3 classical models as benchmarks, and tried to explore the better preprocessing schemes through substitution experiments. This will be elaborated in the experimental section.

C. Model Selection

In our research, we mainly selected 5 classical Machine Learning Algorithm for evaluation, trying to figure out the best suitable model for solving the problem. We will describe the characteristics of each of these nine models and how they can be applied to the solution of this dichotomous problem.

Logistic Regression: Logistic Regression is a statistical method that models the probability of a binary outcome based on one or more predictor variables [5]. Though it is a regression model, we can still use it to solve the binary classification problem by predicting whether the possibility of the result is 1 larger than the threshold.

K-Nearest Neighbors: K-Nearest Neighbors (KNN) is a classical instance-based learning algorithm used for classification and regression [8]. KNN classifies a data point based on the majority class among its k-nearest neighbors in the feature space. The model is easily for adoption in solving our classification problem.

Decision Tree: Decision Tree algorithms are used for both classification and regression tasks. Decision trees split the data at each node based on the feature that provides the maximum information gain or minimum Gini impurity, leading to a tree structure where each leaf represents a class label. It is suitable in handling discrete values and roubust for missing property.

Support Vector Machine: Support Vector Machine (SVM) is a supervised learning model used for classification and regression tasks. SVM finds the hyperplane that best separates the data points of different classes in the feature space.

XGBoost: XGBoost (Extreme Gradient Boosting) is an efficient and scalable implementation of gradient boosting machines. XGBoost includes features such as regularization, parallel processing, and handling missing values, making it highly effective for large-scale and complex datasets. [3]

D. Optimality Analysis

The machine learning algorithm cannot ensure the optimality of the output solution. Instead, the model will tries to minimize the differences between prediction and actual results. However, some of the factors will determine the effectiveness of the machine learning model, which will be discussed below:

1) Dataset Quality: Since the ML model learning prediction policy from dataset. The quality of the dataset can significantly influence the final performance. Unbalanced and biased dataset might misguide the machine learning model in producing wrong prediction, while dataset with duplicate columns might also lead to the overfitting and low robustness of the model. Also, since the dataset might contains missing values which cannot handled by ML models, it is important to preprocess the data input to ensure the model output meaningful prediction.

2) Hyper-parameters: Hyper-parameters play a crucial role in the performance and effectiveness of machine learning models. Unlike model parameters, which are learned from the training data, hyperparameters are set before the training process begins and control the behavior of the training algorithm. The selection on hyper-parameters will significantly influenced the model performance. Thus, we adopt the Grid Search strategy in deciding the best Hyper-parameters

IV. Experiments

In this segment, we aim to evaluate the performance of our preprocessing approaches and model performance mentionedIII. Our experiments will primarily focus on: (a) Comparing various preprocessing Approaches, (b) Evaluating model performance and conduct selection.

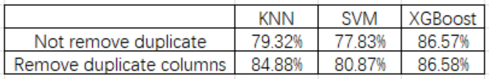
A. Setup

Configuration:
CPU: 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz
GPU: NVIDIA GeForce RTX 3060 Laptop GPU
RAM: 16G
Operating System: Windows 11 23H2
Python Environment:
Python Version: 3.9.13
Sklearn version: 1.4.2

B. Results And Analysis

1) Effectiveness on various Preprocessing Approaches: We evaluate the effectiveness of various preprocessing approaches, with selection on 3 baseline model KNN, XGBoost and SVM. Default hyperparameters were chosen for the model, and five rounds of experiments were performed for each group to calculate their average accuracy, and the results of the experiments are as follows:

Experiment 1: Effectiveness of removing duplicating columns



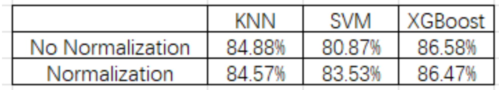
	KNN	SVM	XGBoost
Not remove duplicate	79.32%	77.83%	86.57%
Remove duplicate columns	84.88%	80.87%	86.58%

Fig. 8: Effectiveness of removing duplicating columns

As shown in above fig.8, the operation of removing duplicate rows can significantly improve the performance of KNN and SVM models. This may be due to the fact that the above two models are distance-based models and the reduction of invalid features can effectively improve the prediction accuracy of the models. However, for the XGBoost model, the above improvements do not demonstrate significant performance gains, which to some extent demonstrates the robustness of the XGBoost model for processing complex data.

Experiment 2: Effectiveness of normalization

We further explore the effect of Normalisation on the performance of the model, the steps of the experiment are the same as above and the results are as fig.9



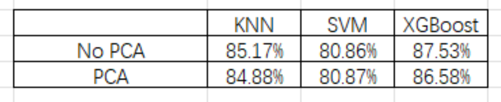
	KNN	SVM	XGBoost
No Normalization	84.88%	80.87%	86.58%
Normalization	84.57%	83.53%	86.47%

Fig. 9: Effectiveness of Normalization

In the above results, the Normalization operation significantly improves the performance of the SVM model, indicating that the model is strongly influenced by the data scale. However, at the same time, the performance of the KNN model and the XGBoost model did not change significantly, demonstrating that these two models are robust to data of different scales.

Experiment 3: Effectiveness of Dimensional Reduction

The experiment follows the above strategy to evaluate the effectiveness of dimensional reduction, with result shown in fig.10



	KNN	SVM	XGBoost
No PCA	85.17%	80.86%	87.53%
PCA	84.88%	80.87%	86.58%

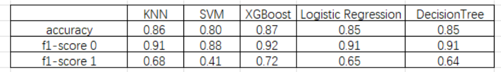
Fig. 10: Effectiveness of Dimensional Reduction

In the above results, the SVM model after the PCA operation has a very slight improvement with the KNN model, while the XGBoost model shows a more significant decrease. This may be due to the fact that the dataset as a whole has too few features with low dimensionality, resulting in the dimensionality reduction operation not being able to effectively remove irrelevant data. At the same time, since dimensionality reduction causes loss of information, it may have affected the overall performance of XGBoost.

2) Effectiveness on Various Machine Learning Models:

Experiment 5: Best performance of each model

We train and select the best hyper-parametric model for the six models mentioned above by means of grid search, and experimentally select the same preprocessing operations. We finally get the model results as follows:



	KNN	SVM	XGBoost	Logistic Regression	DecisionTree
accuracy	0.86	0.80	0.87	0.85	0.85
f1-score 0	0.91	0.88	0.92	0.91	0.91
f1-score 1	0.68	0.41	0.72	0.65	0.64

Fig. 11: Best Performance of different Models

As shown on the graph, most of the models except the SVM model performed relatively well, with an accuracy of more than 85%.The reason for the poor performance of the SVM model may be due to the fact that we did not choose normalisation for uniform preprocessing (an operation that may affect the results of the other models). However, we note that the f1-score gap between the different models is more pronounced, with most models having lower f1-score scores, suggesting that they are affected by an unbalanced dataset and have poor prediction rates for populations with income over 50K.

Finally, we select the XGBoost model as the final predictor model, which has the best accuracy and f1-score, with robustness in handling complex dataset.

V. Conclusion

In this paper, I explore the usage various machine learning model in solving Revenue Prediction Problem. I also explore various approaches in data preprocessing, with analyze of the optimality and characteristic of each model. Additionally, I conduct empirical studies evaluate the performance of preprocessing operations and models. The experimental results show that most of the model perform well under effective data preprocessing, while XGBoost model demonstrate the best performance and roubustness, which is selected for the final output model.

Due to time limitation, our research did not evaluate other classical classification model like Random Forest and AdaBoost, which might be effective in solving this classification problem. Also, further exploration and evaluation on preprocessing approaches can be conducted.

In this project, I delved into machine learning algorithm and finally produce an effective model through comparative attempts, which significantly improved my understanding and knowledge of machine learning. Meanwhile, I enhanced my understanding of python and programming, and how to use libraries like pandas and sklearn. These will benefited me a lot in further study.

VI. Acknowledgement

I would like to express my gratitude to Professor Bo Yuan for his exceptional course. Additionally, I am grateful for the guidance provided by SA in the labs. I truly appreciate the opportunity that CS311 has given me to implement this project. The report may be a little long as it contains a lot of pictures and tables, thanks for your patience for reading until the end.

References

[1] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] N. Chakrabarty and S. Biswas. A statistical approach to adult census income level prediction. pages 207–212, 10 2018.

[3] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[4] V. Chockalingam, S. Shah, and R. P. Shaw. Income classification using adult census data (cse 258 assignment 2). 2017.

[5] J. Cramer. The origins of logistic regression. Technical Report 02119, Tinbergen Institute, 2002.

[6] F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. *arXiv preprint arXiv:2108.04884*, 2021.

[7] T. Evgeniou and M. Pontil. Support vector machines: Theory and applications. volume 2049, pages 249–257, 09 2001.

[8] E. Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. Technical report, USAF School of Aviation Medicine, Randolph Field, Texas, 1951. Archived (PDF) from the original on September 26, 2020.

[9] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[10] N. Naheed, M. Shaheen, S. A. Khan, M. Alawairdhi, and M. A. Khan. Importance of features selection, attributes selection, challenges and future directions for medical imaging data: a review. *Computer Modeling in Engineering & Sciences*, 125(1):314–344, 2020.

[11] K. Potdar, T. Pardawala, and C. Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9, 10 2017.

[12] M. Pérez-Ortiz, S. Jiménez-Fernández, P. A. Gutiérrez, E. Alexandre, C. Martinez, and S. Salcedo-Sanz. A review of classification problems and algorithms in renewable energy applications. *Energies*, 9:607, 08 2016.

[13] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[14] M. Walia, B. Tierney, and S. Mckeever. Synthesising tabular data using wasserstein conditional gans with gradient penalty (wcgan-gp). 12 2020.