Handling Class Imbalance with oversampling for predicting annual income: Duplication vs. synthetic data

Manar Attar (2631465), Majd Al Ali (2659280), Imane Akhyar (2667107), Duaa Ashtar (2696234) and **Raihan Karim Ishmam (2694796)**

Abstract

According to Pew Research Center, 71% of the world population has low or poor income(1). This huge imbalance in income distribution makes it hard for machine learning algorithms to make accurate predictions. In this paper, we decided to train different classifiers using an imbalanced dataset and optimize performance by attempting to balance the data. The aim is to predict the income, for which we use a dataset from the U.S. Census Bureau(2). To deal with the imbalance, two variants of oversampling: random oversampling (duplication) and synthetic oversampling (SMOTE) are compared to see which method performs better for different classifiers. the comparison is done by examining performance metrics such as f1-score and roc-AUC score. We observe an improvement for these scores using both oversampling methods, with slight advantage when using SMOTE-oversampling.

Keywords: *Class Imbalance, Predicting Income, Deep Learning, Classification Algorithms, Machine Learning Models, Decision Trees, Random Forest, Support Vector Machine, Naive Bayes, K-Nearest Neighbours, Logistic Regression*

1 Introduction

1.1 Motivation

One of the significant elements of our daily lives is our financial activity and status. From travel to food to shelter, all are highly dependent on and run by our finances (payments, rents, bills, etc.). Moreover, the most common source of providing this financial support is their income if we focus on adults. This makes income prediction an area of interest for many organizations and researchers. One of the most reliable datasets that we can use, including the data relevant to the income, is the census dataset. To find a dataset of this type, we turned to two very well-known web repositories: Kaggle and the UCI - Machine Learning Repository . We initially aimed to find our datasets from the ongoing competitions in Kaggle, but we could not find an appropriate one matching our project. So we moved to the UCI repository, where we found a data set that contains weighted census data extracted from the 1994 and 1995 population surveys conducted by the U.S. Census Bureau(2). The data set contains demographic and employment-related variables, which would provide us with a set of reasonably relevant features.

1.2 Paper outline

The essential element of this project is using machine learning to predict the $annual$ income of the various instances (adults). We decided to approach the problem as a classification problem rather than a regression one, as this would let us focus more on the statistical aspects of the performance. Choosing regression might have led to more detailed predicted values for the income, but that might introduce issues with the overfitting when we try to improve the accuracy, so we opted to go with classification. The machine learning classifier that we are developing is designed to do binary classification, again due to the reasons above. We chose to go with a specified value of (50,000 USD) to split the annual income range into two classes, so the instances are classified as, with an annual income either greater than 50,000 USD or less.

The dataset that we are using is comparatively significant, with around 32,500 instances. We are planning to use several different training models like K - Nearest Neighbours, Logistic Regression,

Handling Class Imbalance with oversampling for predicting annual income: Duplication vs. synthetic data

Name1 Surname1 (ID1), Name2 Surname2 (ID2), Name3 Surname3 (ID3)
Name4 Surname4 (ID4) and Name5 Surname5 (ID5)

Abstract

Abstract text here.

Keywords: keyword1, keyword2, keyword3, keyword4, keyword5, keyword6, keyword7, keyword8, keyword9, keyword10

1 Introduction

1.1 Motivation

Motivation text here.

2 Data Exploration

Data exploration text here.

3 Research Question

Research question text here.

4 Data preprocessing

4.1 Missing values

Missing values text here.

4.2 Feature selection

Feature selection text here.

5 Methodology

5.1 K-Nearest-neighbor (KNN)

KNN text here.

5.2 Decision trees

Decision trees text here.

5.3 Random forest

Random forest text here.

5.4 Logistic regression

Logistic regression text here.

5.5 Support vector machines (SVMs)

SVM text here.

6 Results

6.1 Hyperparameter optimization results

Hyperparameter optimization results text here.

7 Discussion

Discussion text here.

8 Conclusion

Conclusion text here.

References

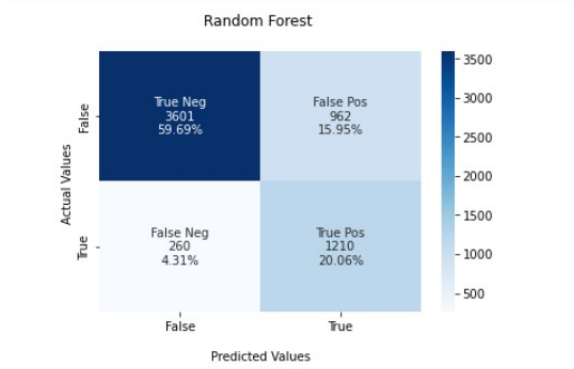
[1] Reference 1.

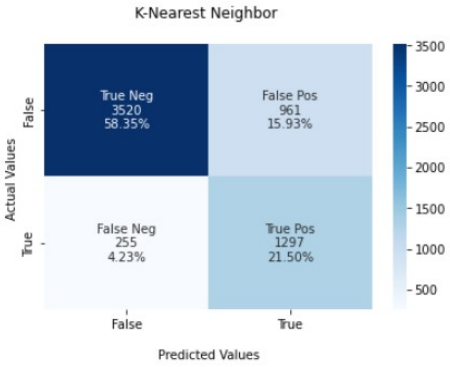
[2] Reference 2.

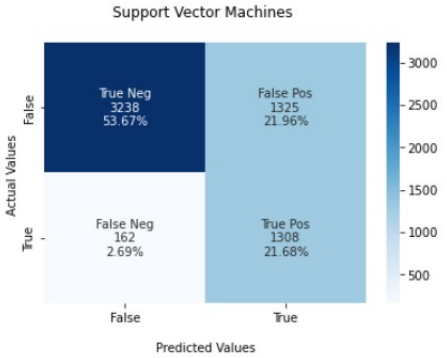
[3] Reference 3.

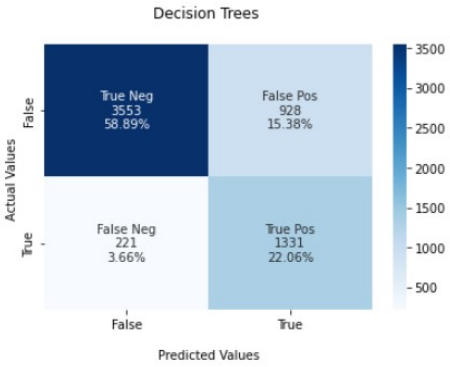
Appendix

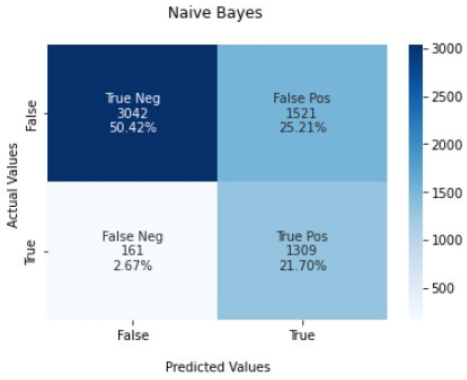
Appendix text here.











Appendix 1: Confusion matrices (Random Oversampling)