Census Income Prediction using SVM

Introduction

Census income prediction is the process of estimating a person's income level using demographic data. This information is frequently utilised in a range of applications, including credit scoring and targeted advertising. The UCI Census Income dataset, an open-source dataset that contains details on people from the 1994 Census Bureau database, is one of the most well-liked datasets for this endeavor.

Each instance in the UCI Census Income dataset has 15 variables, including age, education, and employment, as well as a binary label indicating whether the person's annual income is $50,000 or less.

For the purpose of predicting census income, machine learning models can be trained and evaluated using this dataset. Decision trees, random forests, and gradient boosting machines are examples of frequently used algorithms. Accuracy, precision, recall, and F1-score are some of the measures that are frequently used to measure the performance of trained models.

Data Exploration

First of all, the data was analyzed and explored to get some useful information from the dataset. Dataset was having 32560 records and each record was having 15 values means there are 15 columns in the dataset. 9 features were having object data type and 6 were of numeric type. Then I used different visualization to analyze the data and get useful insights from it that are given below.

- 75 percent people were having less than 50K income.
- Around 70 percent people works in private sector according to given data.
- Most people were high school graduate.
- 67 percent data were for male.
- 46 percent people were married and were living with spouse.
- Executive managers are getting high income and professors speciality by Occupation
- Husbands were earning more income than singles.

All insights have a plot that can be seen in coding results.

Data Cleaning

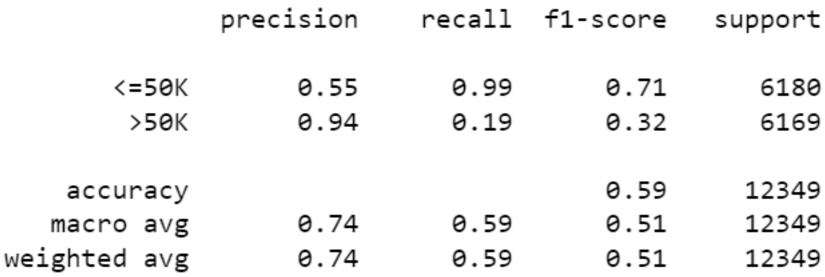
Data cleaning is the process of transforming raw data into some useable format so that machine learning algorithms can get trained on such data and could be used to make the prediction in real world. So, in this project I also clean data and in cleaning first of all I check the null values but luckily there are no null values in the whole data and then I checked duplicated records and I found that there are 24 duplicated records in the whole data set that were dropped.

After that I encoded categorical features into some numerical values because machine learning algorithms can only work with numbers They cannot work with string values or with categorical values. So, I encoded categorical features using label encoder from sklearn Library.

Then after that I checked the distribution of the target feature and came to know that there are a greater number of records for one class which belongs to people that were earning less than 50K. And for other class I was having a smaller number of records so if we will train our model on such data the model will get a biased towards the class that is having a greater number of records. So, that's what we don't want. We want to have a fair model and for that I balance the data using over sampling technique and for that I used SMOTE from imblearn library.

Modeling

Before applying model directly on the data, first I split my data into train and test part by the ratio of 75:25 percent. Sport Vector Machine algorithm was assigned to use in this assignment so I trained it on the training data and evaluated it on testing data. I achieved an accuracy of 60% on the test data. The classification report of the model is even below.

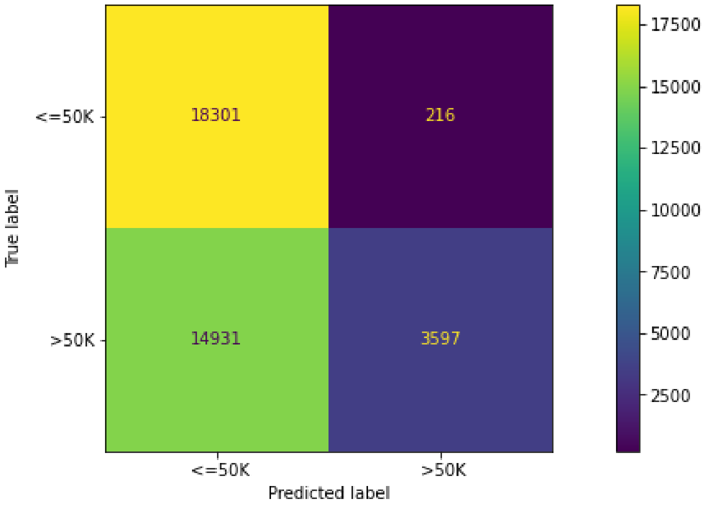


```
              precision    recall  f1-score   support

       <=50K       0.55      0.99      0.71      6180
        >50K       0.94      0.19      0.32      6169

    accuracy                           0.59     12349
   macro avg       0.74      0.59      0.51     12349
weighted avg       0.74      0.59      0.51     12349
```

From this we can see that model has become biased towards class having salary less than 50K even we have balanced the data. This is a little bit low accuracy because model becomes biased. The confusion matrix of model is given below.


<=50K
>50K
True label
<=50K
>50K
Predicted label