Santiago Guada

Prof Jing Gao

ECE 49595: Data Mining

Final Project: Predicting Adult Income Based on Personal Information

Introduction

The problem I'm trying to solve with this project is helping people understand the reason of their income. In today's day we have a big problem of people not knowing or understanding the reason why they are not getting promoted or earning higher income. In my opinion, personal finance and income increase is one of the most important things in having a job, and with this study we can show the different aspects of a person that help with this. I believe that the impact it can have for people to understand what are the characteristics of their profile that they can improve will bring more people to get promotions and higher income from their job and other businesses.

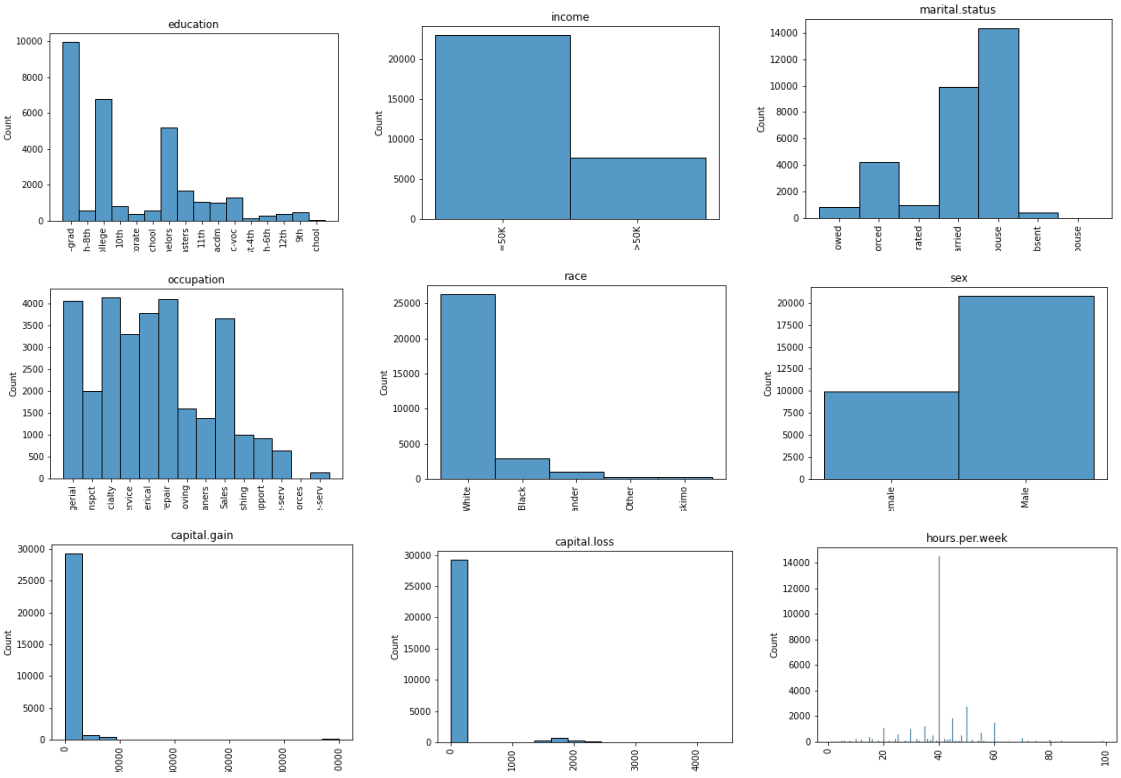
Formulation

The data mining task that can be formulated is a classification task since the income recorded in the data is "<=50K" or ">50K", therefore we can use classification to predict our results. The input of the model will be age, education, marital status, occupation, race, sex, capital gain, capital loss, and hours per week of a person surveyed and the output will be if the person's income is below or above 50K a year.

Datasets

I got the dataset from Kaggle (https://www.kaggle.com/datasets/uciml/adult-census-income?resource=download). I preprocessed the data by first eliminating the columns I found unnecessary or redundant for the analysis to eliminate analysis that is non-essential to what I was looking for. Then, I eliminated the rows missing information. After this, I performed the exploratory data analysis by plotting a histogram of all the attributes of the data to understand more and start to make some assumptions of the data. Finally, I separated the data into train and test, and transformed all the non-numerical data into numerical data by assigning values to the different representations, trying to make the expecting higher values that could impact the result the most as the higher values. In here, I realized that the data doesn't have a fair representation of the attributes, for example, there are about twice the number of men that there are of women. Another interesting observation of the data can be seen on the education attribute, where most of the people surveyed (~30%) got High School degree as their highest level of education, roughly 20% didn't finish High School, and about 50% have some college degree or higher. On the marital status part, we see that most of the people either got married or not, we see a minority of people divorced, widowed, etc. In terms of race, we see a majority of about 85% that are white. For the labels, we see that about 75% of the people surveyed have an income of less than 50K. For the correlation graph below, we see that for the income section, the biggest correlation attributes are marital status, education, and age. These results are understandable because usually people that are married have combined income or have planned to expand the amount of people their income is covering. For education, we see a high expectation that as the years of education increase, the person is more specialized in a topic that is usually rewarded highly in industry, and with age the older

you are the more experience the person has. Below we see the EDA graphs containing histograms, correlation map, and pivot table.



Algorithm

I applied SVM, linear SVC, and K nearest neighbor's classification algorithms. Got this inspiration from scikit-learn cheat sheet https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html .

Experiments

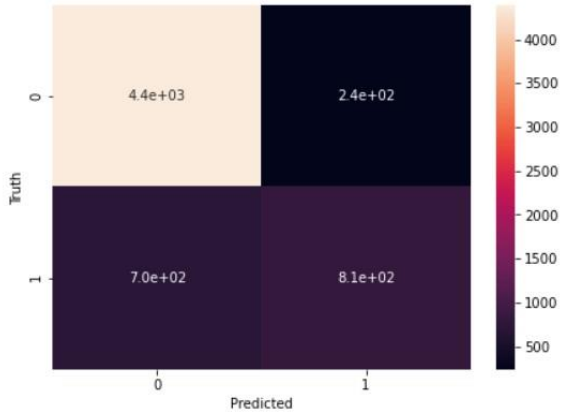
To evaluate the output, I calculated the accuracy, precision, and recall with the predicted results of each algorithm used. Here are the snapshots of the result:

SVM

```
Accuracy: 0.8465169270833334
Precision calc: 0.8616290480863592
Recall calc: 0.9485738980121002
```

```
from sklearn.metrics import confusion_matrix
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
cm_svm = confusion_matrix(y_test, y_pred_svm)
plt.figure(figsize=(7,5))
sn.heatmap(cm_svm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(42.0, 0.5, 'Truth')
```

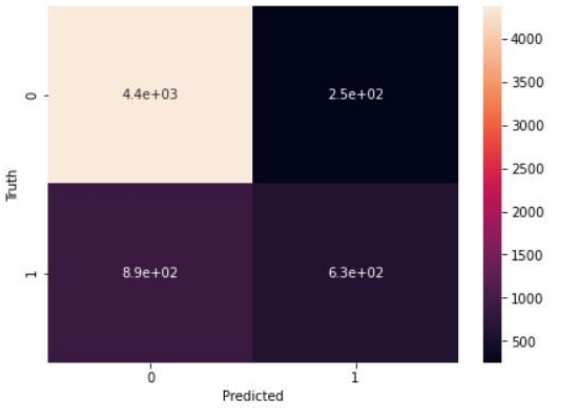


KNN

```
Accuracy: 0.8146158854166666
Precision calc: 0.8312132143535219
Recall calc: 0.945980985306828
```

```
from sklearn.metrics import confusion_matrix
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
cm_knn = confusion_matrix(y_test, y_pred_knn)
plt.figure(figsize=(7,5))
sn.heatmap(cm_knn, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(42.0, 0.5, 'Truth')
```

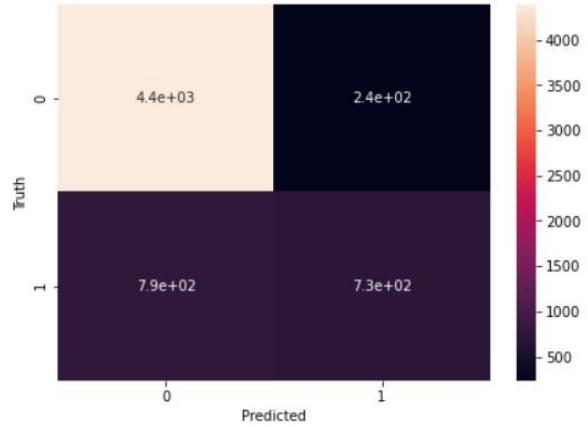


Linear SVC

```
Accuracy: 0.8336588541666666
Precision calc: 0.8482039397450754
Recall calc: 0.9490060501296457
```

```
from sklearn.metrics import confusion_matrix
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
cm_svc = confusion_matrix(y_test, y_pred_svc)
plt.figure(figsize=(7,5))
sn.heatmap(cm_svc, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(42.0, 0.5, 'Truth')
```



Comparison

The performance of the different algorithms I tested are similar between each other, having an average accuracy of about 83%. I believe that since the result of this dataset is binary, using binary classification is the best way to go here and these algorithms are good for this task. Another algorithm that could be implemented is Logistic regression. I also performed some external comparison since in Kaggle people also post their result, and the average accuracy I saw in other people's result was about 86%, so it can be concluded that my models are working well on predicting the results.

Challenges

Some challenges in the data were on deciding which attributes to take out and which ones to leave in. However, I believe that the attributes I decided to take were good and concise between them. Another challenge was converting the non-numerical values to numerical because it was tedious to do this for every column.

GitHub Repository

https://github.com/sguadav/SalaryClassification