DOCUMENTATION

MACHINE LEARNING

GEN 511

CENSUS INCOME PREDICTION

TEAM MEMBERS :
I.Cherish (IMT2017022) cherish.chowdary@iiitb.org
S.Sanjay (IMT2017037) sanjaykumar.reddy@iiitb.org
S.Prasanth (IMT2017525) sivaramannagari.prasanth@iiitb.org

24th November, 2019

Contents

1 Abstract

For this hackathon, we examine the Census Income Dataset available at MLdata. We aim to predict whether an invidual's income is greater than $50,000 per year based on several attributes from the consensus data.

2 Problem Statement

The data contains anonymous information such as age, occupation, education, working class, etc. The goal is to train a binary classifier to predict the income which has two possible values >50K and <50K. There are 48842 instances and 15 attributes in the dataset.
The dataset has the following attributes:

- age: Represents the age of the individual.
- workclass: Represents the employment status of the individual.
- fnlweight: Final sampling weight. This is the weight of each data entry i.e. this is the number of people each data entry represents.
- education: Highest level of education completed by the individual.
- education_num: 'education' represented in numerical form.
- marital_status: Represents the marital status of the individual.
- ocupation: Type of ocupation of the individual.
- relationship: Represents how this individual is related to others. Each entry only has one relationship attribute.
- race: Describes an individual's race.
- sex: Gender of the individual.
- capital_gain: Capital gains for an individual.
- capital_loss: Capital loss for an individual.
- hours_per_week: Hours an individual has worked in a week.
- native_country: Country of origin of the individual.
- income_level: Whether an individual makes more than $50,000 annually.

3 Exploratory Data Analysis

3.1 Data Visualization

Firstly, we assigned 0 to the label <50k and 1 to >50k. To gain insights about which features would be most helpful for the model, we looked at the attributes and the distribution of entries of income_level.

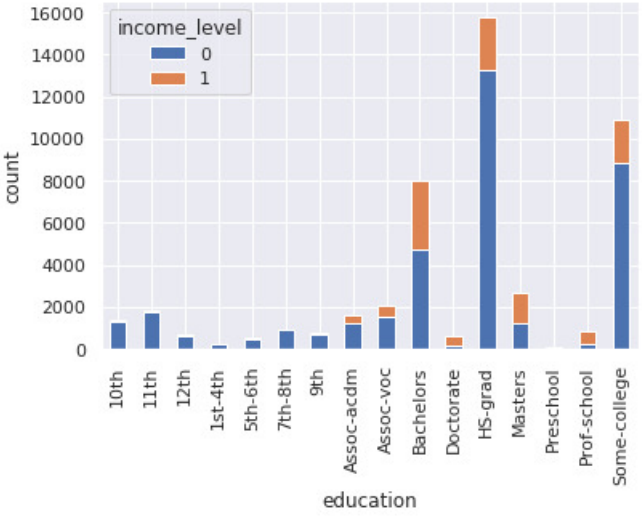

Figure 1: Education vs Income

Figure 2: WorkClass vs Income

Figure 1 shows the distribution of the different levels of education among individuals in the dataset. Most of the individuals in the dataset have at most a high school education while only a small portion have a doctorate. It also shows the relationship between education and income. For the most part, a higher level of education is correlated to a higher percentage of individuals with the label >50k.
As we can see from Figure 3, the majority of individuals work in the private sector. The one concerning statistic is the number of individuals with an unknown(?) work class. The probability of earning greater than $50,000 annually is almost the same for all classes except for federal government and self-emp-inc.

As seen in Figure 3, Occupations are uniformly distributed in the dataset. Looking at the income distribution, exec-managerial and prof-speciality have a higher percentage of individuals earning over $50,000. On the other hand, the percentage of individuals earning the same is significantly lower. The graph also shows the higher number of individuals with unknown occupations.

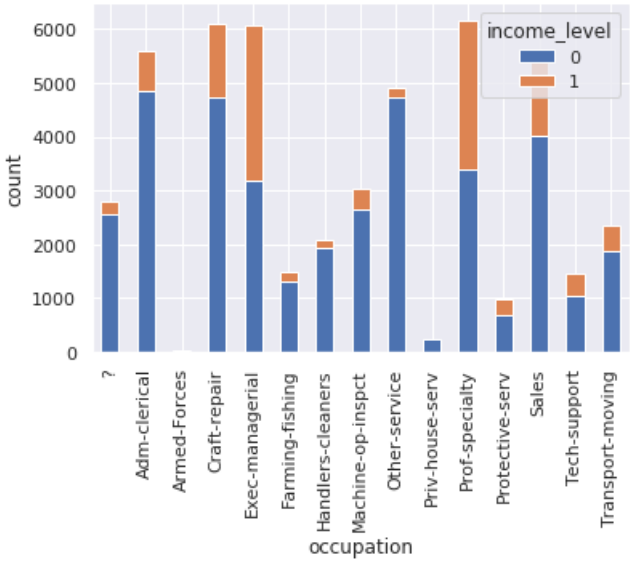

Figure 3: Occupation vs Income

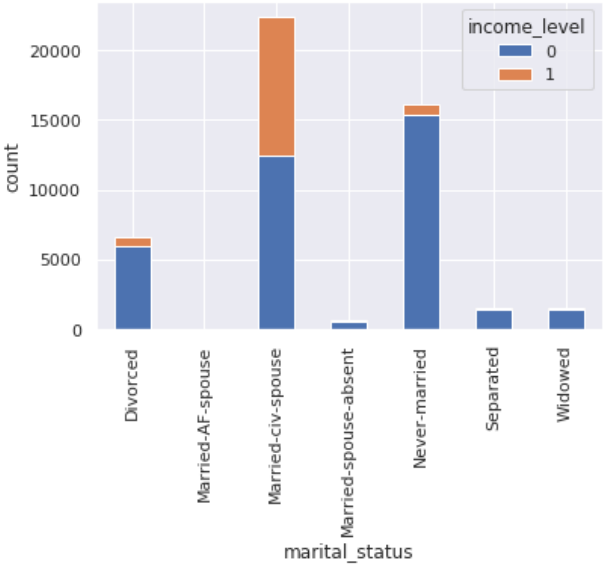

Figure 4: Marital Status vs Income

Marital status describes the marital status of the individual. It might appear that marital status does not relate to income, but looking at Figure 4, we can see that married-civ-spouse has very high percentage of individuals with income greater than $50,000. Initially, we considered leaving this attribute, but after looking at the graph we decided that we should consider it in our model.

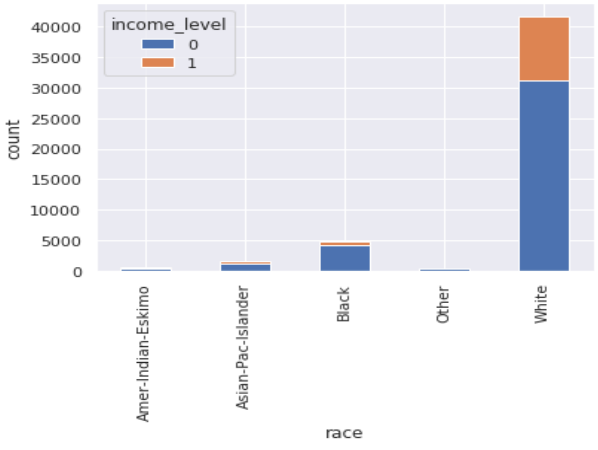

Figure 5: Race vs Income

As we can see from Figure 5, the sample size of white is proportionately larger than the rest of races. Whites and Asians have a larger percentage of entries greater than $50,000 than the rest of the races.

From Figure 6, we can see that there is almost double the sample size of males in

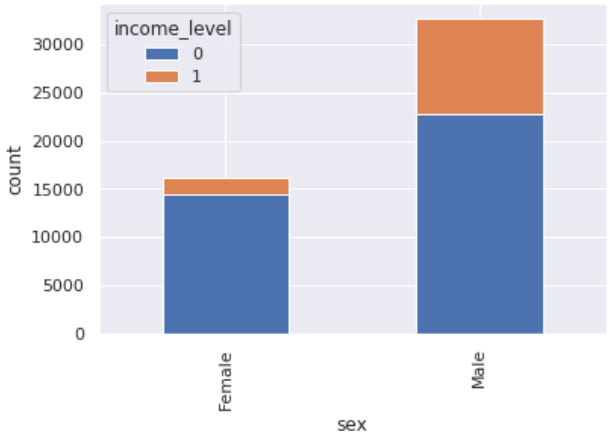

Figure 6: Sex vs Income

comparison to females in the dataset. While this may not affect our predictions too much, the distribution of income can. The percentage of males who make greater than $50,000 is much greater than the percentage of females that make the same amount. This will certainly be a significant factor, and should be a feature considered in our prediction model. Below is the correlation matrix of the features.

	age	workclass	education	occupation	race	sex	hours_per_week	native_country
age	1	0.0858917	-0.00351071	-0.00447381	0.0237763	0.0820533	0.101992	-0.00317959
workclass	0.0858917	1	0.0178462	0.0175084	0.0497648	0.0696384	0.0513659	0.00388862
education	-0.00351071	0.0178462	1	-0.0334994	0.0114566	-0.0275686	0.0608872	0.077893
occupation	-0.00447381	0.0175084	-0.0334994	1	-2.58868e-05	0.0566247	0.0161596	-0.0028484
race	0.0237763	0.0497648	0.0114566	-2.58868e-05	1	0.0889348	0.0447381	0.124342
sex	0.0820533	0.0696384	-0.0275686	0.0566247	0.0889348	1	0.231425	-0.00417012
hours_per_week	0.101992	0.0513659	0.0608872	0.0161596	0.0447381	0.231425	1	0.00755399
native_country	-0.00317959	0.00388862	0.077893	-0.0028484	0.124342	-0.00417012	0.00755399	1

Figure 7: Correlation Matrix

3.2 Dealing with Nulls

The data had three features with unknown values in them. Firstly, we replaced all the unknown values with NaNs. Then replaced them with the following values:

- work_class: Replaced the nulls with the mode of the data(converted into numbers using label encoding). The mode of the data is 'Private'.

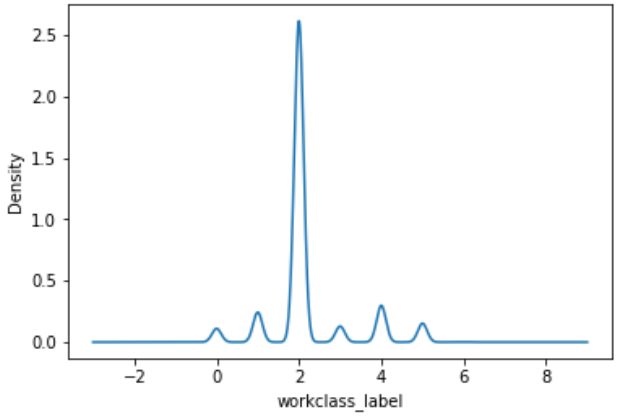

Figure 8: Work Class Distribution

- native_country: Replaced the nulls with the mode of the data(converted into numbers using label encoding). The mode of the data is 'United-States'.

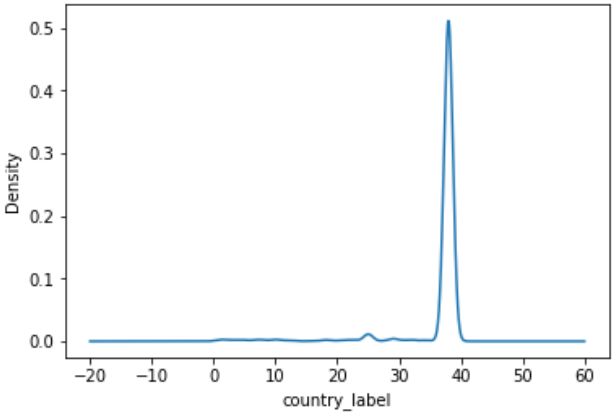

Figure 9: Native Country Distribution

- occupation: As we can see from Figure 10, the graph is not gaussian. So we used logistic regression to predict it using the features age, work_class, education, race, sex, hours_per_week, native_country, income_level. We trained a model using these features, since they are highly related to occupation, and replaced the null values in occupation with their corresponding values from the model.

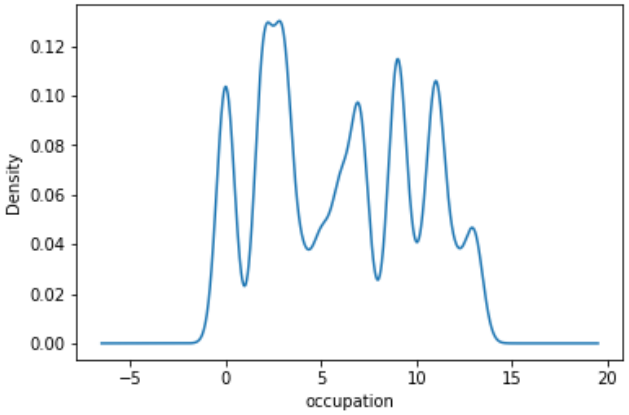

Figure 10: Occupation Distribution

3.3 Removal of Attributes

We opted not to use the feature education_num, since this feature is highly correlated to education and did not contribute for our analysis.

3.4 Normalising Features

We normalized the data as part of preprocessing. Our goal was to change the values of numeric columns to a common scale, without distorting the differences in the ranges of values. The features age, fnlwgt, capital_gain, capital_loss, hours_per_week were normalized.

3.5 One Hot Encoding

Most of the features in our data are categorical i.e. they contain labels. Since most machine learning algorithms cannot work on labels, we had to convert them into numerical form. We had two options: Label Encoding and One Hot Encoding. Using Label encoding may result in poor performance because of the natural ordering

between categories. So we used One Hot Encoding on work_class, education, marital_status, occupation, relationship, race, sex, native_country. We ended up getting 105 columns in our dataset. We used these as features for our model.

4 Model

We divided the dataset into two parts: train and test in 70:30 ratio.
Since this is a classification problem, we tried using models like Logistic regression, Random Forest, Decision Tree and XGBoost for predicting the labels. After doing the preprocessing and once the data set is ready, we calculated the accuracy of the corresponding models. The below table and Figure 11 show the accuracies of the models in testing data set.

Model Used	**Accuracy**
Logistic Regression	85.13%
Decision Tree	81.50%
XGBoost	86.30%
Random Forest	85.41%

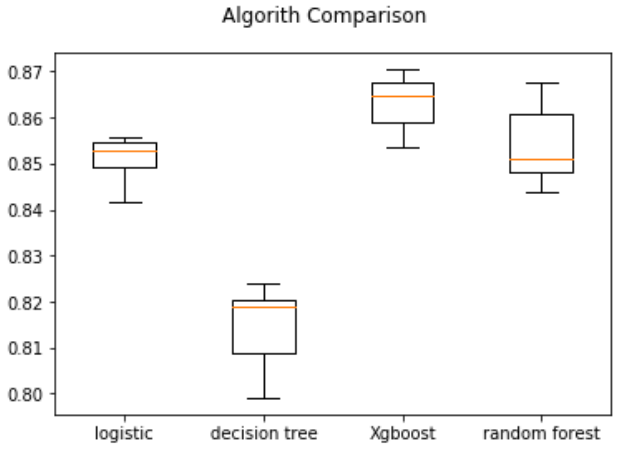

Figure 11: Box Plot of accuracies

As we can see, XGBoost gave the best accuracy score followed by Random Forest. Decision Tree gave least accuracy. But a model should not be considered 'best' solely on its accuracy. So we checked the precision, recall and f1-scores of the models. The below tables show the same.

Logistic Regression			
	Precision	**Recall**	**F1-score**
0	0.88	0.93	0.91
1	0.73	0.61	0.67
macro average	0.81	0.77	0.79

XGBoost			
	Precision	**Recall**	**F1-score**
0	0.84	0.97	0.90
1	0.83	0.40	0.54
macro average	0.83	0.69	0.72

Random Forest			
	Precision	**Recall**	**F1-score**
0	0.89	0.93	0.90
1	0.72	0.62	0.67
macro average	0.80	0.77	0.79

As we can see, XGBoost has the least recall and f1-score in predicting the label, >50k(1). So finally, after calculating the macro average and analysing the data, we decided that Random forest gave the best possible model with accuracy **85.41%** over the testing set.

5 Links to the data and pickle files

1.Data:
https://drive.google.com/file/d/1qmt-WUGpkfPd7ni$_X$9$RZVS8D6WEvGH52/view?usp = sharing$

2.Model Object:
https://drive.google.com/file/d/1cQdTdwcBeYYoIUaqjErK6rwFXW$_f w711/view?usp = sharing$

6 References

1.https://www.mldata.io/datasets

2. https://towardsdatascience.com/attack-toxic-comments-kaggle-competition-using-fast-ai-b9eb61509e79

3.https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python

4.https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python