Predicting and Analyzing Census Income

Thong Tran
ttran156@student.gsu.edu

Hoa Pham
hoa-pham@gmail.com

Department of Computer Science, Georgia State University

Table of Content

Abstract:

In this research paper, we are going to present techniques in data mining/ machine learning that we use to discover and analyze the hidden interesting patterns of Adult Census Income from UCI machine learning repository. In addition, we are also be able to perform a prediction classification task and recommendations/rules from apriori algorithm. Furthermore, we will propose our new algorithm to find or filter the best rules/recommendations among thousands of different rules. Lastly, we will also introduce our new way to handle missing values as a future reference work.

Introduction:

Nowadays, many people have always dream of earning as many income as possible. As a result, people usually make a plan of steps in order to achieve that goal. For instance, people might want to earn higher degree by taking master or even PhD in the exchange of getting a better job and of course a better pay. However, life might change slightly depend on different aspects such as marriage, number of hours they work, or even which country they work for. In other words, having a higher degree does not guarantee that a person can earn more income. As a result, this is a motivation for research team to discover or mine in details of the data and transfer to meaningful information. The aim is to discover what a person should know or have a better plan in order to earn more income. We experimented various machine learning models to estimate the accuracy of the prediction task and succeed at 87%. We also performed a recommendation task algorithm to recommend for a person, who earns less than 50k, about what they should change in order to make more than 50k dollars.

Research Aims:

Our research aims are to predict and recommend a person what they should do if they earn less than 50 thousands dollars per year. In addition, we want to know deeper about what attributes play an important role in deciding the classification result. Lastly, we also want to propose a new way to handle missing values.

Previous Research:

There are some previous research for census income in the hope of identifying the best machine learning models to perform prediction task. Here are the few research papers pros and cons.

- [1] uses various techniques to identify the most importance features and perform prediction task based on Logistic Regression. However, it did not go deeper into finding the best models.
- [2] and [3] performs the same steps as [1] with more findings on machine learning models. However, their chosen models are not considered the "best" due to the fact that the data is not linear and there are some biases that existed.
- [4] is actually really good research project where it aims to find the strongest feature and use association rules to find the correlation/association between different attributes in the data.

Most of the projects above are actually impressive upon finding the best attributes associated and construct different models. However, they are not reliable on predicting with the highest accuracy among top 4 project is 83%. Our hope is to find the next best model to acquire the best accuracy as well as finding an algorithm to find the best meaningful recommendation for a person who is going to use our product.

Data Preprocessing:

Before we want to do anything with the data we got, we need to make it a completed data. In real life, data is not always completed before handed to engineers or scientists. Therefore, computer scientists invented many ways to fill out these missing data. Since, we have not yet thought of any better way to fill out the missing data, so we tried different algorithms and test which one is the best.

Information about data:

We got data from UCI machine learning repository. Our data set contains 32561 rows with 14 attributes each row. There are 2399 rows with missing data:

14 Attributes: Age, workclass, fnlwgt, education, educationnum, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native country, income.

Filling missing data methods:

Here are the methods that we tried:

- Delete row that has missing data
- Replace the missing value with average of that particular attribute

- Replace the missing value with the most frequent value in that particular attribute
- Use support vector machine to predict missing values (future implementation)

After filling missing data with those methods, we test the accuracy by using 9 fold cross validation. It appears that the Mode frequent method was the best candidate for missing value. Finally, we decided using it to do the job.

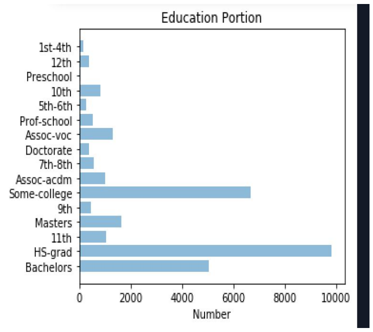
Encoding Data

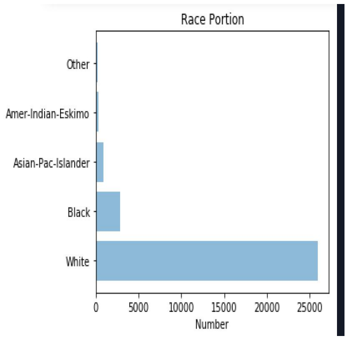
One of the problem in applied machine learning model or data mining techniques is that some machine learning algorithms can recognize and interpret with categorized data such as Decision Trees. However, there are some algorithms that cannot operate with categorized data labels. As a result, for the purpose of this research, we should not limit the number of algorithms used for this project, so we need to convert all the categorized data into numeric data.
We will use LabelEncoder library in Sklearn as a tool to help us tackle down this task

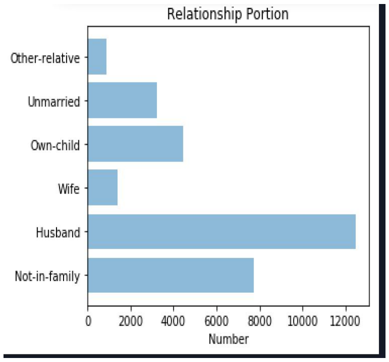
Data Visualization:

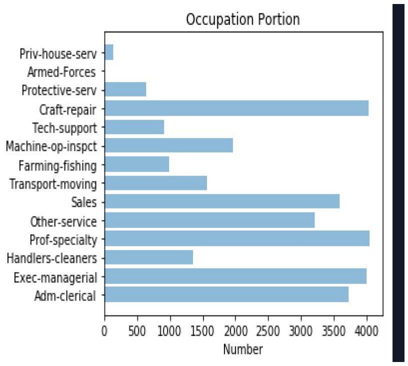
In order to understand and make productive model, we need to investigate how this data is represented.

1. **Distribution Graph**









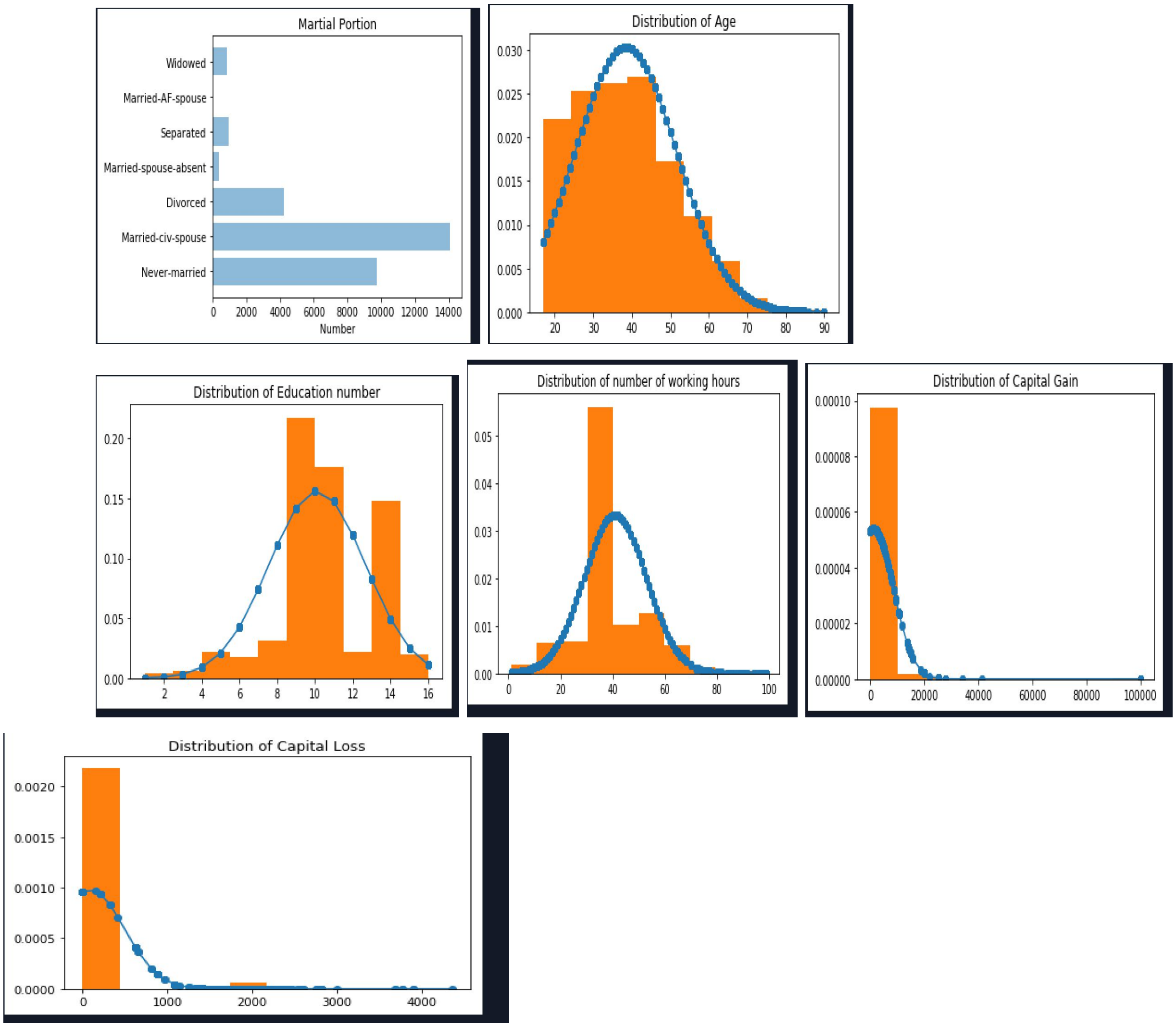
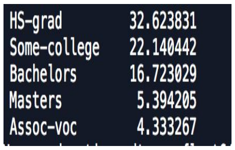
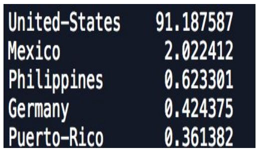


Figure 1: Graph Contribution of Education, Race, Relationship, Occupation, and Marital

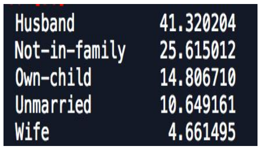
In figure 1, we observed the following statistic values (in percentage)



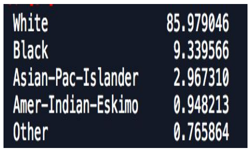
```
HS-grad         32.623831
Some-college    22.140442
Bachelors       16.723029
Masters          5.394205
Assoc-voc        4.333267
```



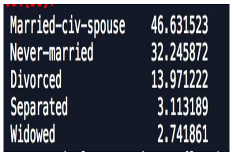
```
United-States   91.187587
Mexico           2.022412
Philippines      0.623301
Germany          0.424375
Puerto-Rico      0.361382
```



```
Husband         41.320204
Not-in-family   25.615012
Own-child       14.806710
Unmarried       10.649161
Wife             4.661495
```



```
White                 85.979046
Black                  9.339566
Asian-Pac-Islander     2.967310
Amer-Indian-Eskimo     0.948213
Other                  0.765864
```



```
Married-civ-spouse    46.631523
Never-married         32.245872
Divorced              13.971222
Separated              3.113189
Widowed                2.741861
```

For continuous data that we collected, we can see that person who was high school graduation, a husband, married, white and live in the US, obtained the highest portion in our data.

From our point of view, we have found the correlation between education and number of years in education. This makes sense because the higher degree the person has, the number of years that person spent for education increases as well. In order to demonstrate the correlation between different independent variables in this data. We have the following correlation graph.

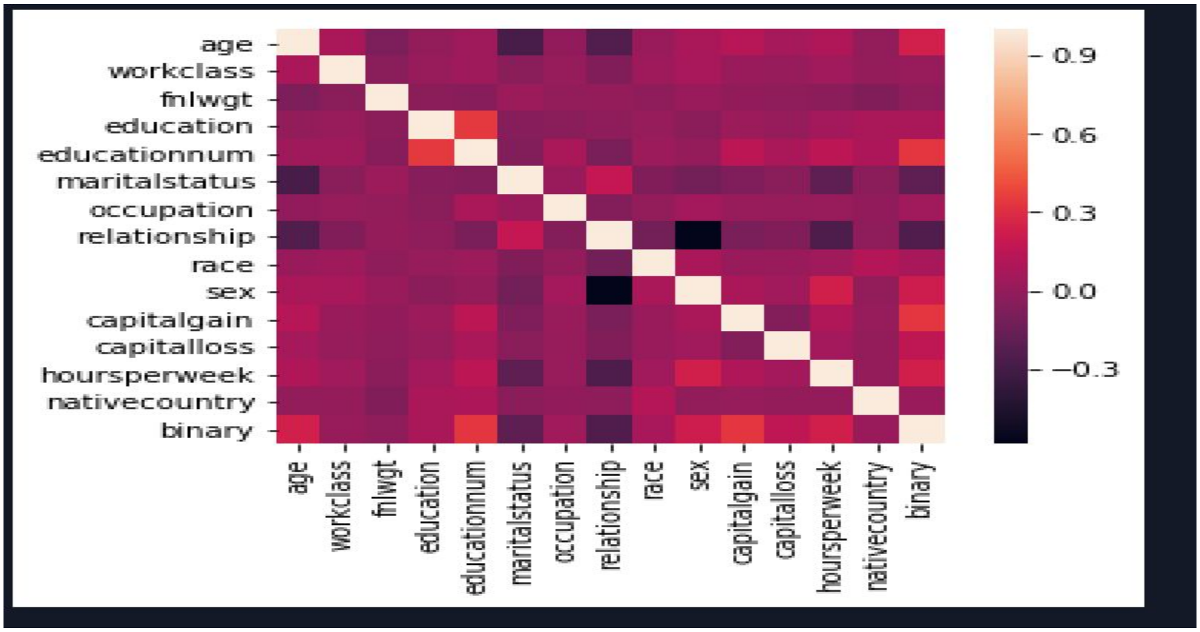
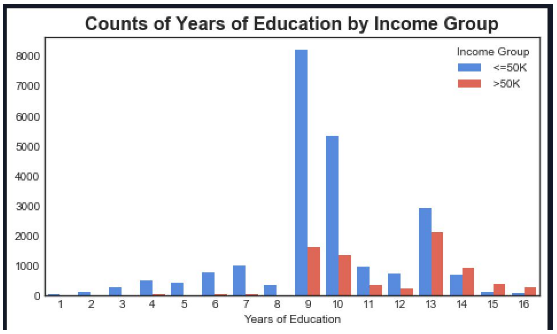
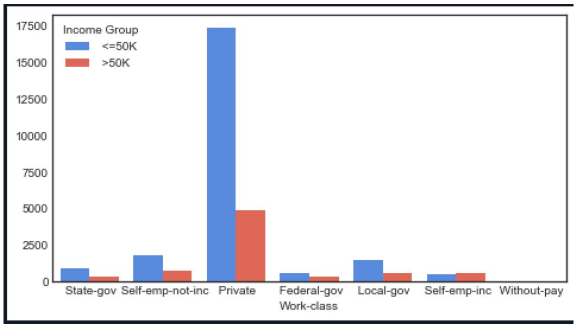


Figure 2: Correlation graph between independent variables

As we can see, the number of years in education correlates with education degree. As a result, that would be reasonable to remove a education degree from the model since the number of years in education can represent someone's degree.

However, Let's do further investigation about the data between two group of income which is group of income more than 50k and group of income less than 50k. Psychologically, we strongly believe that the number of hours working, number of years in education, the workclass, and the marital status would positively impact to the outcome. Let's see how that could be tacked





Construct Model Prediction

In order to perform the best performance for our models, we will eliminate all the data points that has missing values.

Feature Selection:

One of the important technique in data mining to get a good or better accuracy whilst requiring less data and increase the performance time is feature selection or attributes selection where it will include or exclude the attributes from the data for the machine learning training method without changing the data.
We have used three machine learning method to help us tackle this task:

- We performed Boosting algorithm using XGBoost. Here are the results: Capital_Gain(1),Age(2), Years of Education(3), hoursperweek(4), occupation(5), relationship(6), workclass(7)
- We performed Extra Tree algorithm which is a variant for random forest tree. Here are the results: Occupation(1), yearsOfEducation(2), age(3), race(4), marital-status(5), sex(6)
- We performed Random Forest. Here are the results: Age(1), education(2), sex(3), occupation(4), marital-status (5), workclass(6)
- We performed RFE logistic regression. Here are the results: Sex(1), years of education(2), marital-status(3), relationship(4), race(5), work-class(6)

As mentioned above, we removed the degree attribute due to the fact that it has strong correlation with years of education attribute. In addition, marital-status and relationship also have the correlation but not as strong as degree and years of education, as a result, we keep it. We also have noticed that 93% of data points has United States as a native country which do not contribute nominal distribution. As a result, we remove it as well. In short, we tried several subset of attributes, and the promising subset attributes that can fit into the training model are Capital-Gain, Age, Years of Education, Occupation, hoursperweek, marital-status, relationship, and work-class

Machine Learning Models:

We have performed various of machine learning models and here is the short summary table of all the models that we have used

Machine Learning Models	Results
SVM	75.92%
9 Fold Cross Validation	81.047%
Multi Layer Perceptron	82.67%

XG Boosting	86.167%
KNN	80.0139%
Extra Tree Classifier	82.67%
Random Forest (100 Sub Tree)	81.173%

As you can see in the table above, XG Boosting takes even higher accuracy. This is because Boosting helps transform weak learners to strong learners. By that mean, weak learner is a classifier that is not well correlated with true classification, whereas true learner is well correlated.

As a result, we will use Boosting algorithm as a tool to make a prediction whether or not they will earn more than 50k. In the next section, we will more focus on Association Rules and our proposed algorithm to find the best rules among different association rules between different attributes.

Association Rules:

Meaningful Attributes:

Choosing meaningful attributes is important because this will be an information to present to our users and by that mean, all attributes that are presented must be readable and understandable for common human sense. In order to complete this task, I will loop through the attributes and eliminate all unnecessary attributes

I have removed following attributes:

1. In this case, since final weight is really a numeric attribute and I do not understand what it is, so I can assume that others would not understand it as well. Therefore, final weight is removed
2. Capital Gain and Capital Loss is removed, but they are not meaningful for association rules task
3. Degree has the correlation with years of Education;therefore, years of education is sufficient enough to represent the type of the degree that a person has

We also step further by analyzing other numeric attributes such as age, hoursperweek, years of education. All of these attributes are ranged differently, so we need to categorize them into a block of range.

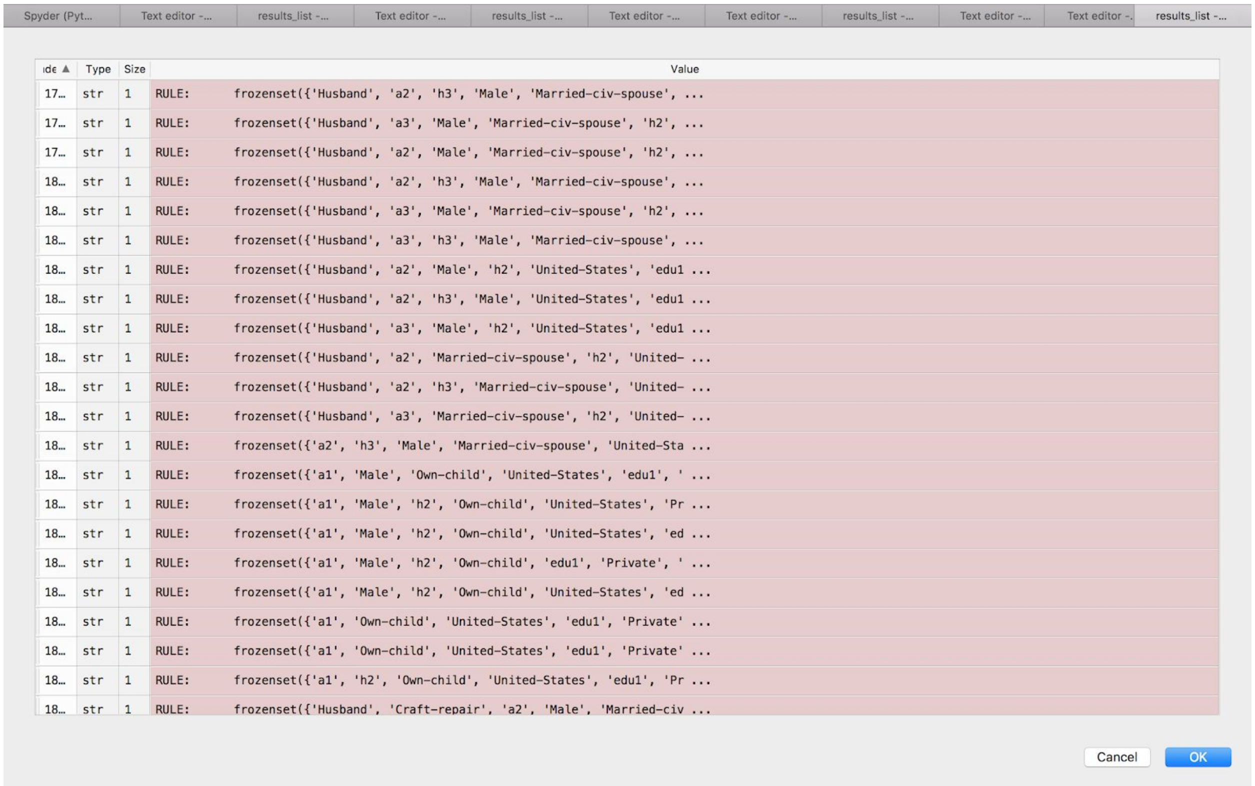
Age: a1: 0 to 25, a2: 25 to 40, a3: 40 to 60, a4 for >60

Years of Education: E1: 1 to 12, E2: 12 to 14

Hours Per Week: H1: 0 to 20, H2: 20 to 40, H3 for > 40

Apriori Algorithm

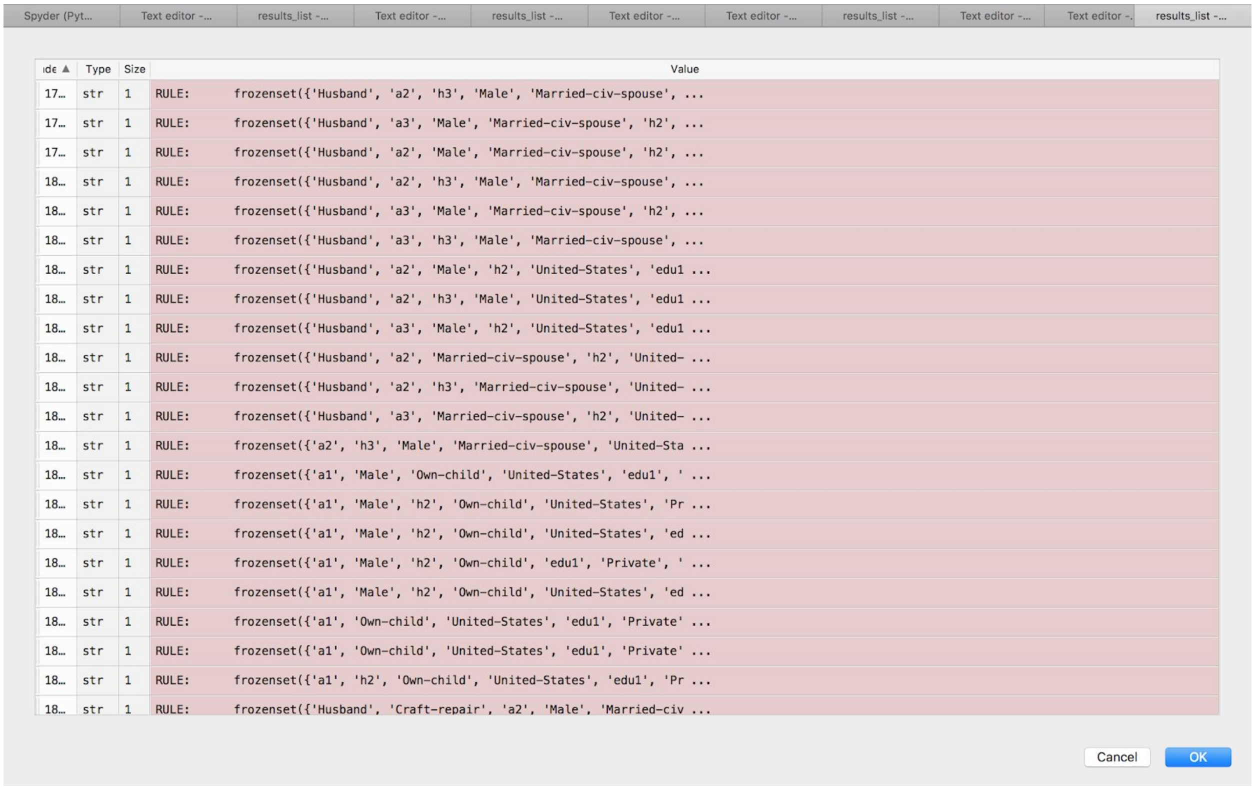
We have used Apriori Algorithm from Python library. Here is the screenshot result



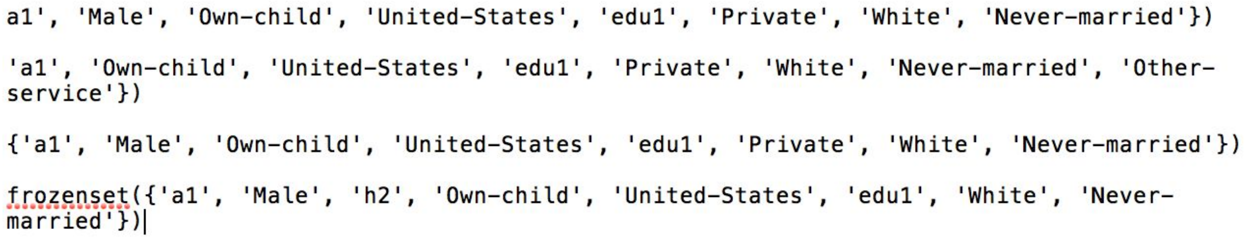
ide ▲	Type	Size	Value
17...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Male', 'Married-civ-spouse', ...
17...	str	1	RULE: frozenset({'Husband', 'a3', 'Male', 'Married-civ-spouse', 'h2', ...
17...	str	1	RULE: frozenset({'Husband', 'a2', 'Male', 'Married-civ-spouse', 'h2', ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Male', 'Married-civ-spouse', ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'Male', 'Married-civ-spouse', 'h2', ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'h3', 'Male', 'Married-civ-spouse', ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'Male', 'h2', 'United-States', 'edu1 ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Male', 'United-States', 'edu1 ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'Male', 'h2', 'United-States', 'edu1 ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'Married-civ-spouse', 'h2', 'United- ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Married-civ-spouse', 'United- ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'Married-civ-spouse', 'h2', 'United- ...
18...	str	1	RULE: frozenset({'a2', 'h3', 'Male', 'Married-civ-spouse', 'United-Sta ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'Own-child', 'United-States', 'edu1', ' ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'Pr ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'ed ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'edu1', 'Private', ' ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'ed ...
18...	str	1	RULE: frozenset({'a1', 'Own-child', 'United-States', 'edu1', 'Private' ...
18...	str	1	RULE: frozenset({'a1', 'Own-child', 'United-States', 'edu1', 'Private' ...
18...	str	1	RULE: frozenset({'a1', 'h2', 'Own-child', 'United-States', 'edu1', 'Pr ...
18...	str	1	RULE: frozenset({'Husband', 'Craft-repair', 'a2', 'Male', 'Married-civ ...

```
'Husband', 'a3', 'h3', 'Exec-managerial', 'Male', 'Married-civ-spouse', 'United-States',
'edu2', 'Private', 'White'})

'Husband', 'a2', 'h3', 'Exec-managerial', 'Male', 'Married-civ-spouse', 'United-States',
'edu2', 'Private', 'White'})

'Husband', 'a3', 'Male', 'Married-civ-spouse', 'h2', 'United-States', 'edu2', 'Private',
'White'})
```

Figure 3: Result Rules for people that have more than 50k



ıde ▲	Type	Size	Value
17...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Male', 'Married-civ-spouse', ...
17...	str	1	RULE: frozenset({'Husband', 'a3', 'Male', 'Married-civ-spouse', 'h2', ...
17...	str	1	RULE: frozenset({'Husband', 'a2', 'Male', 'Married-civ-spouse', 'h2', ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Male', 'Married-civ-spouse', ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'Male', 'Married-civ-spouse', 'h2', ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'h3', 'Male', 'Married-civ-spouse', ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'Male', 'h2', 'United-States', 'edu1 ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Male', 'United-States', 'edu1 ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'Male', 'h2', 'United-States', 'edu1 ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'Married-civ-spouse', 'h2', 'United- ...
18...	str	1	RULE: frozenset({'Husband', 'a2', 'h3', 'Married-civ-spouse', 'United- ...
18...	str	1	RULE: frozenset({'Husband', 'a3', 'Married-civ-spouse', 'h2', 'United- ...
18...	str	1	RULE: frozenset({'a2', 'h3', 'Male', 'Married-civ-spouse', 'United-Sta ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'Own-child', 'United-States', 'edu1', ' ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'Pr ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'ed ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'edu1', 'Private', ' ...
18...	str	1	RULE: frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'ed ...
18...	str	1	RULE: frozenset({'a1', 'Own-child', 'United-States', 'edu1', 'Private' ...
18...	str	1	RULE: frozenset({'a1', 'Own-child', 'United-States', 'edu1', 'Private' ...
18...	str	1	RULE: frozenset({'a1', 'h2', 'Own-child', 'United-States', 'edu1', 'Pr ...
18...	str	1	RULE: frozenset({'Husband', 'Craft-repair', 'a2', 'Male', 'Married-civ ...



```
a1', 'Male', 'Own-child', 'United-States', 'edu1', 'Private', 'White', 'Never-married'})

'a1', 'Own-child', 'United-States', 'edu1', 'Private', 'White', 'Never-married', 'Other-
service'})

{'a1', 'Male', 'Own-child', 'United-States', 'edu1', 'Private', 'White', 'Never-married'})

frozenset({'a1', 'Male', 'h2', 'Own-child', 'United-States', 'edu1', 'White', 'Never-
married'})
```

Figure 4: Result Rules for people that earn less than 50k

We have collected the results for both people that earn more than and less than 50k income with minimum support count 60% and minimum confidence as 90%. We have almost 5 thousand of different rules and some of them are strongly correlated. From there, we can have a better understanding about this data, and that is:

- *People that earn more than 50k*: are the people whom their age is more than 50, they have very long term of being in school, they are usually a male and married, they also work as 40 hours or more per week and they also live in the US
- *People that earn less than 50k:* are the people whom their age is from 20 to 40, they do not have long term in school (Most of them have high school degree), they are usually male and not married, and they also do not have family relationship.

As a result, we need to filter all the best rules among all of these rules in order to provide a better information. In a section below, we are going to discuss our new algorithm to filter the best rules

Our New Algorithm (H-T Algorithm)

After using Boosting algorithm to verify if a person could earn less than 50k. We are going to use H-T algorithm to filter the rules for that person.
We noticed that in order to recommend the rules for a person, all attributes must be changeable. In other words, the attribute such as Sex, Race will not be included from a recommendation dataset because they are not changeable. We can’t recommend someone to change their gender or their race to earn more money.

First Attempt

Our first attempt is to count similarity between the user’s information to all different rules, and return the “rules” that has the highest similar to the user.
For example, let’s denote u1 is the user input
u1 = ["Male","a1","h1","edu1", "United-States", "Sales", "Never-married"]
This person is young, but he has quite a bit long term education. He is also working less than 40 hours per week and lives in the United-States.
Boosting algorithm indicates that this person earns less than 50k.

Algorithm:

1. Apply Association Rules for the people who earn more than 50k only. Filter all the rules that have the length of items more than 6, and exclude “Race” and “Sex” out of the rules
2. Loop through all the rules and count the similarity with user’s input

Stimulation Result

['Husband', 'edu2', 'a3', 'United-States', 'Sales', 'Married-civ-spouse']
['Husband', 'edu2', 'h3', 'a2', 'Prof-specialty', 'Married-civ-spouse']
['Husband', 'Private', 'a2', 'United-States', 'Craft-repair', 'Married-civ-spouse', 'edu1']
['Husband', 'Private', 'a3', 'United-States', 'Craft-repair', 'Married-civ-spouse', 'edu1']
['Husband', 'Male', 'a3', 'United-States', 'Married-civ-spouse', 'Exec-managerial', 'edu1']

['Husband', 'Male', 'a3', 'United-States', 'Married-civ-spouse', 'Exec-managerial', 'h2']
['Husband', 'Male', 'edu2', 'a2', 'United-States', 'Married-civ-spouse', 'Exec-managerial']
['Male', 'edu2', 'Private', 'a3', 'United-States', 'Married-civ-spouse', 'Exec-managerial']
['Husband', 'Male', 'edu2', 'Private', 'a2', 'Prof-specialty', 'United-States', 'Married-civ-spouse']

The rules we received have 795 out of 4499 rules, and the table above is a subset of 795 rules However, this result is not quite "best" because 795 rules are still quite long and we need to be able to filter it down furthermore. In addition, due to the fact of counting similarity between different attributes, we have mistakenly selected "silly" rules that are bound around the user's input. As a result, the rules will be generated with fewer differences. From our perspective, we need to be able to handle different rules based on the "nomination". For example, if a person has short term education, then the H-T algorithm should intelligently recommend the person to study higher, or it can recommend the person to change their current job to another if that is suitable based on their age, their marital-status and so on. That is the reason why our first attempt fail because it is not considered all different aspects that I mentioned above. As a result, we will reveal our second attempt in the hope of getting better rules

Second Attempt (in progress)

There are attributes that cannot be changed such as sex, race, relationship. We cannot recommend any attributes that user cannot change. Also, there are two attributes that affect our rules the most which is hours of work and years of education.

The new attempt is to keep all rules that has unchangeable attributes these attributes will be get from user's input. Then, output all the rules that have better "hour of work" attribute (h2 or h3) and and "year of education" (edu2).

Algorithms' steps:

1. Looping through array of attributes getting from user's input. Filter out rules that do not have unchangeable attributes

2. Splitting all rules into three groups

    Group 1: All rules has edu1 and edu2

    Group 2: All rules has h1, h2 and h3

    Group 3: The rest

3. Looping through user's input

    If there is a "edu1" outputs all the rules that has "edu2" or "edu3"

    If there is a "h1", outputs all the rules that has "h2" or "h3"

If education and hour of work was missing in the user’s input, outputs all rules based on unchangeable attributes

Let use the same example as the first attempt:
u1 = ["Male","a1","h1","edu1", "United-States", "Sales", "Never-married"]

Stimulation Result:

['Husband', 'Male', 'Private', 'h3', 'United-States', 'Married-civ-spouse']
['Husband', 'Male', 'Private', 'h2', 'Married-civ-spouse', 'White']
['Husband', 'Male', 'Private', 'h3', 'Married-civ-spouse', 'White']
['Husband', 'Male', 'United-States', 'h2', 'Married-civ-spouse', 'White']
['Husband', 'Male', 'h3', 'United-States', 'Married-civ-spouse', 'White']
['Husband', 'Male', 'Private', 'h3', 'United-States', 'White']
['Husband', 'Male', 'Private', 'h3', 'United-States', 'White']
['Husband', 'Private', 'h3', 'United-States', 'Married-civ-spouse', 'White']
['Male', 'Private', 'h3', 'United-States', 'Married-civ-spouse', 'White']
['Husband', 'Male', 'Private', 'h3', 'United-States', 'Married-civ-spouse', 'White']
['Husband', 'Male', 'edu2', 'United-States', 'Married-civ-spouse', 'White']

This algorithm is not efficient, it takes O(n^3). We will review and improve the performance of this algorithm.

Proposed Solution for Solving Missing Values:

Missing value is a bug, and this section will be introduced our new proposed solution to handle missing values.
If the dataset is supervised learning, we will suggest to use SVM to solve this type of problem.

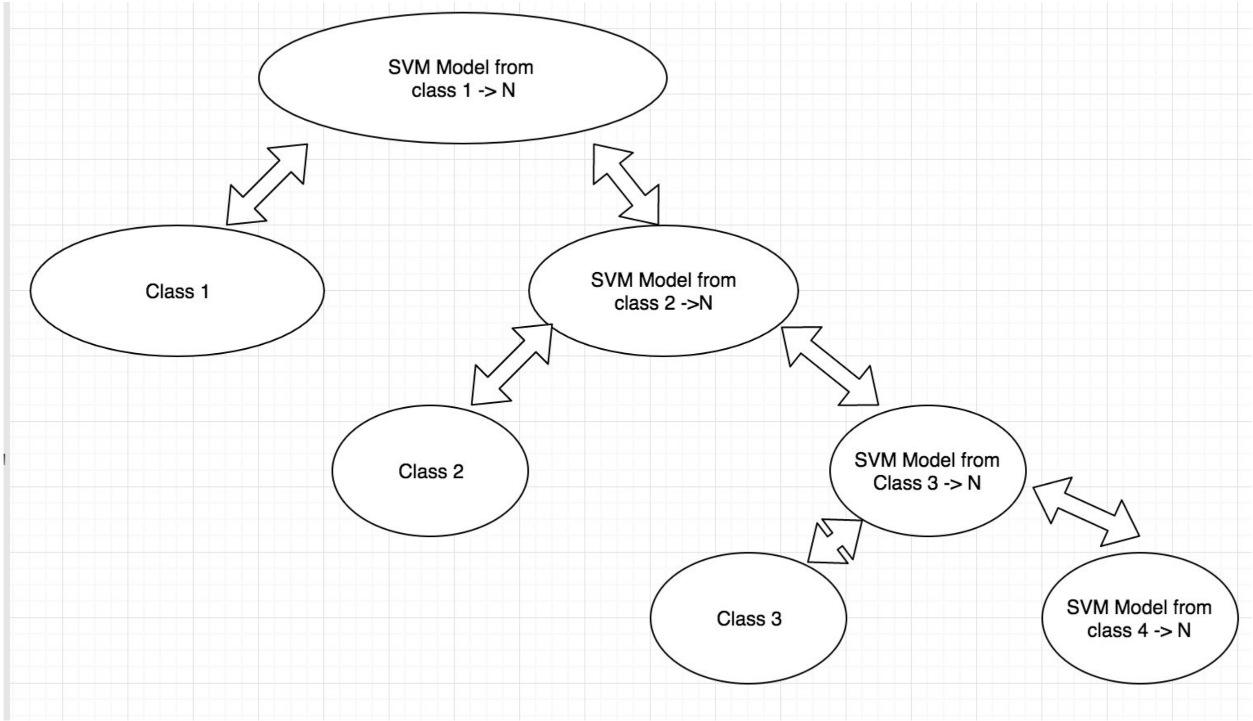
Algorithm Steps:

1. Filter all the data into two different major subset. First subset would be the data that has no missing values, and the second subset has missing values

2. Filter the second subset into many different small subsets. Each subset will contain different attribute where it has missing values. For example, if attribute A and B has missing values. We will separate them into two different subset where subset A only contains missing values for attribute A and the same as subset B
3. For the first subset that has complete data points, we will then separate them into different attributes as well. Now, the values in each attribute become a new classification labels.
4. Perform SVM for each training set and missing set based on different attribute and use the prediction to predict the missing values.

Due to the fact that we will due to different classification label for different attribute and the data we have is huge, we can consider this is a big data mining problem. The way to handle is to change our normal SVM to a custom SVM algorithm which is Binary SVM. The advantage of Binary SVM is that we can use it to handle multi-class label with recursive.

In binary SVM, we will then push class label to the left node and recursively on the right node. We will keep doing recursively until all the labels are exhausted. The left and the right are determined by the distance between one subset to every other subset as a whole group.


SVM Model from class 1 -> N
Class 1
SVM Model from class 2 ->N
Class 2
SVM Model from Class 3 -> N
Class 3
SVM Model from class 4 -> N

Proposed Solution for predicting the best rules based on user input:

Apriori algorithm give us 4499 rules to recommend for user, but we simply cannot recommend a random one. Our goal is to predict the most accurately rule based on the information that user input. We figured out the way to do so, but we could not have time to finish. However, we will soon implement this algorithm.
Here is the general ideas of how our algorithm works.

Algorithm:

1 Determine which attributes are unchangeable and which attributes can be changed from user's input.
2 Compare the user inputs with each of the rules we have. While comparing, we filter out the rule does not have all unchangeable attributes that user has.
3 Outputting all rules that we have left.
4 Finding the rules that has the shortest length. Shortest length means the rules have least attributes (the more attributes in the rule, the more the user need to change). These are options for user to choose from. We do not want strictly give the user just one option, because it may be impossible for someone to move to another country.

Future Work and Implementation

For future work, we hope to achieve the following mandatory requirements:

1. We hope to complete our proposed solution to fix missing values using Binary SVM to predict better results
2. We hope to complete and create a Restful APIs using Django Web Framework to create a web application and open source to different developers
3. We hope to create an another attempt for an algorithm to filter better recommendation for a person to help them decide what they can do to achieve higher income.
4. While researching H-T algorithm for a second attempt, we also take a look of using Deep Neural Network to solve this problem. Since the outcome that we expect will be rely on the fit training data from the user and output the correct rules. We will also implement our Deep Neural Network algorithm after finishing our H-T algorithm and compare it.

Summary

In summary, we use the most frequent method for filling missing data and in the future we will use svm to fill our missing data. To encode data, we used LabelEncoder library in Sklearn. Then, to see how our data is represented, we plot graphs of Education, Race, Relationship, Occupation, and Marital. Then, to fill out the unimportant attributes from out data we used four algorithm which is random forest, rfe, boosting ,and extra tree algorithm. We construct our prediction model using XG boosting algorithm (accuracy is 86.167%). For recommending for user we used the Apriori Algorithm, and the issue with this is that Apriori recommends to many rules and not accurate. Our first attempt algorithm had successfully filtered out smaller subset rules. However, these rules were still considered “silly” rules by the fact that these rules do not consider all different aspects from the user to provide a better outcome. We decided to develop a new algorithm to filter out these rules, the algorithm is in progress and will be soon publish. Next version of this product is to publish on the website using django framework, so everybody can use it as a guide for their better future.

References

[1] “Census Income.” *Data Mining with R*, individual.utoronto.ca/zabet/census-income.html.

[2] https://cseweb.ucsd.edu/~jmcauley/cse190/reports/sp15/024.pdf , by Jim Cauley

[3] https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a120.pdf , by Vidya Chockalingam, Sejal Shah, and Ronit Shaw

[4]Antonov, Anton Antonov. “Classification and Association Rules for Census Income Data.” *Mathematica for Prediction Algorithms*, 1 Apr. 2016, mathematicaforprediction.wordpress.com/2014/03/30/classification-and-association-rules-for-census-income-data/.

[5] “Sklearn.feature_selection.RFE¶.” *Sklearn.feature_selection.RFE - Scikit-Learn 0.19.1 Documentation*, scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.