Project1

Name: Jiecheng Gu
NetID: jgu13

1 Data Introduction

In this project, I use UCI Adult Census Dataset as the raw data. The source of this dataset is http://archive.ics.uci.edu/ml/datasets/Adult.

As shown in Table1.1, there are 15 attributes in this dataset (adult.data) with 32561 instances. Five attributes including age, fnlclass, education-num, capital-gain, capital-loss and hours-per-week are continuous numeric attributes while other attributes are categorical attributes.

Table1.1 Data types of attributes

Attribute name	Data type	Attribute name	Data type
age	continuous	relationship	categorical
workclass	categorical	race	categorical
fnlclass	continuous	sex	categorical
education	categorical	capital-gain	continuous
education-num	continuous	capital-loss	continuous
marital-status	categorical	hours-per-week	continuous
occupation	categorical	native-country	categorical
class	categorical	/	/

2 Data Preprocessing

There are several steps to process the raw data:

a. Delete the transaction data containing "?"

There are 2399 rows of transaction data containing unknow value "?" in this dataset.

b. Delete the duplicate data

There is no duplicate data in this dataset.

c. Delete the useless columns

The value in fnlclass column is unique in each transaction data but it has nothing to do with the frequent pattern mining, so I delete this column. Furthermore, the education-num column is corresponding to the education column, so they are treated as duplicate columns in this project and I delete the education-num column.

e. Convert numeric columns into categorical columns

There are both numeric and categorical data in this dataset. In order to keep the consistency in mining frequent patterns, I convert numeric columns into categorical columns by grouping them into multiple ranges.

For the age column, I cut it into levels: *Young* (1-25), *Middle-aged* (26-45), *Senior* (46-65) and *Old* (66+).

For the hours-per-week, I cut it into levels: *Part-time* (0-25), *Full-time* (25-40), *Over-time* (40-60) and *Too-much* (60+).

For the capital-gain and capital-loss columns, I cut them into three levels based on their medians: *None* (0), *Low* (greater than 0 and below the median) and *High* (above

the median).
After data preprocessing, there are 13 attributes in this dataset with 30162 instances (Table2.1). Figure2.1 shows the first five rows of data in this dataset after data preprocessing.

Table2.1 Data types of attributes after data preprocessing

Attribute name	Data type	Attribute name	Data type
age	categorical	race	categorical
workclass	categorical	sex	categorical
education	categorical	capital-gain	continuous
marital-status	categorical	capital-loss	continuous
occupation	categorical	hours-per-week	categorical
class	categorical	native-country	categorical
relationship	categorical	/	/

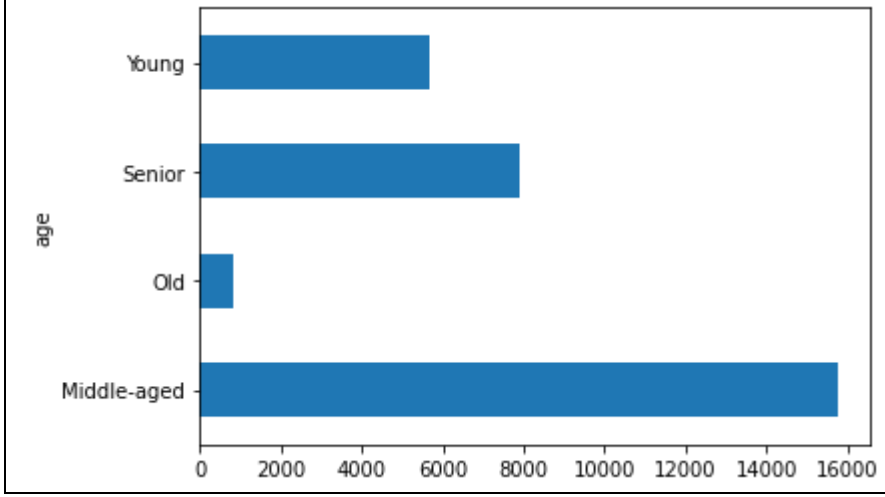
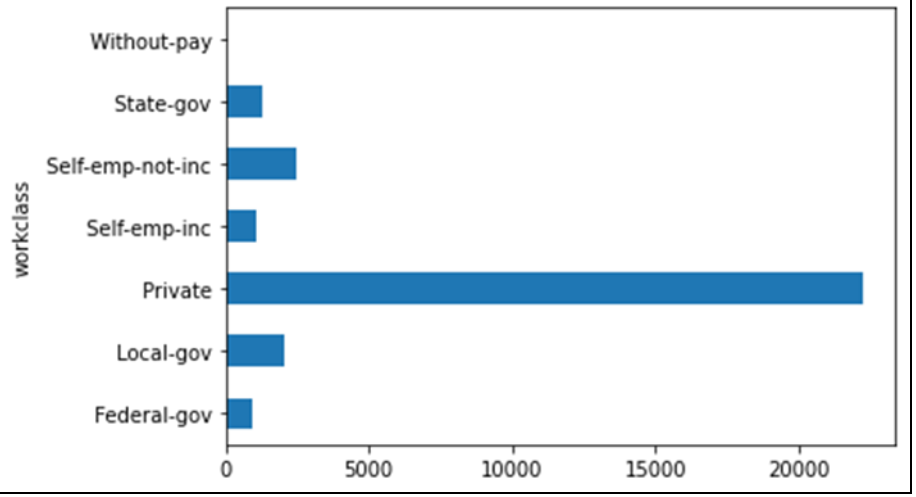
	age	workclass	education	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class
0	Middle-aged	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	High	None	Full-time	United-States	<=50K
1	Senior	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	None	None	Part-time	United-States	<=50K
2	Middle-aged	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	None	None	Full-time	United-States	<=50K
3	Senior	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	None	None	Full-time	United-States	<=50K
4	Middle-aged	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	None	None	Full-time	Cuba	<=50K

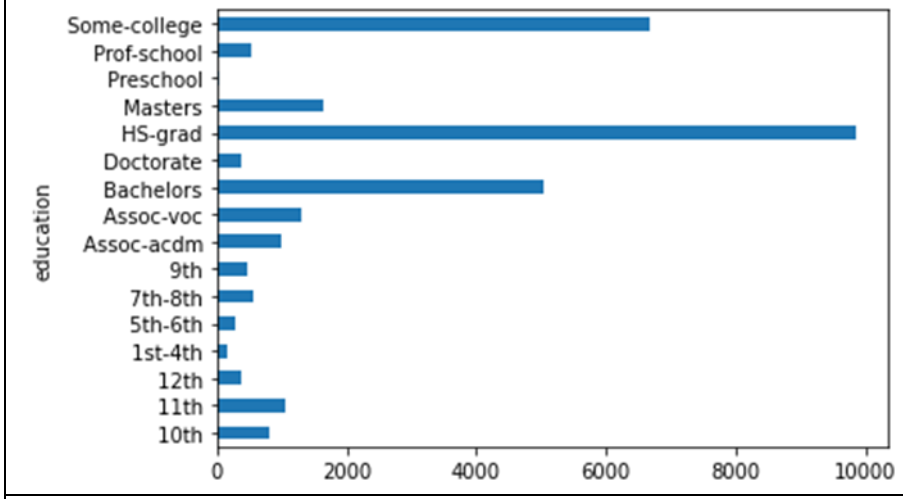
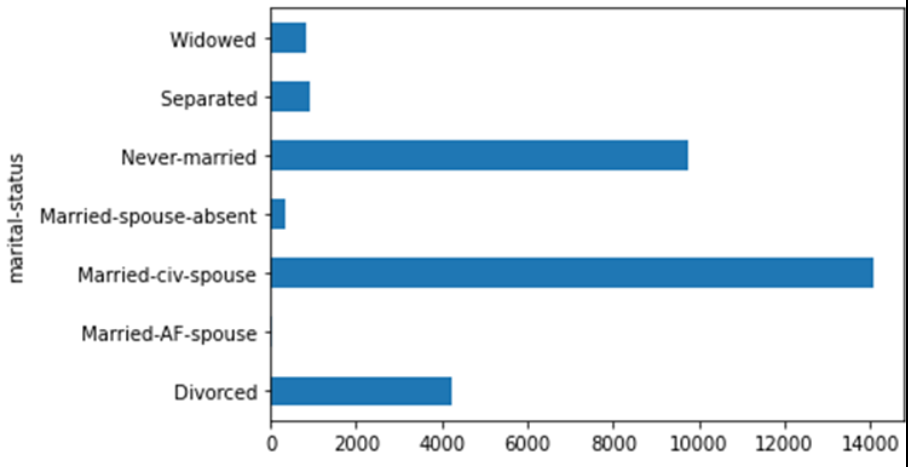
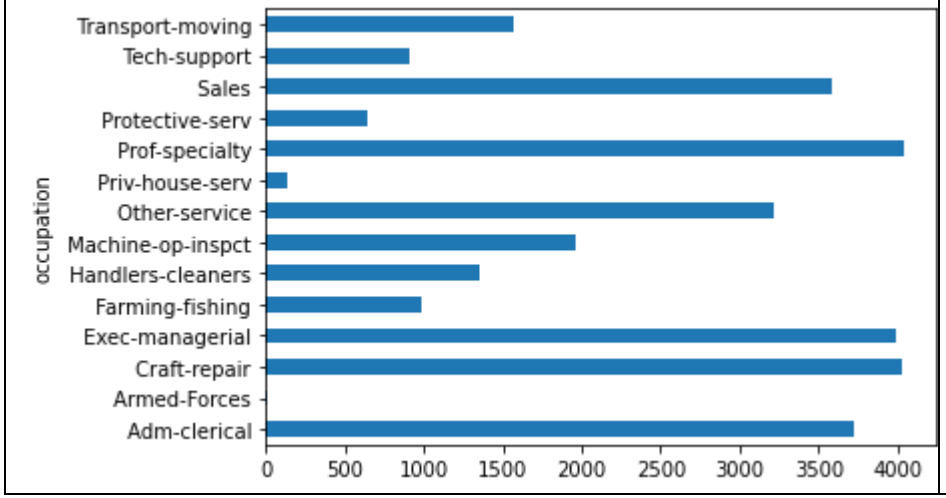
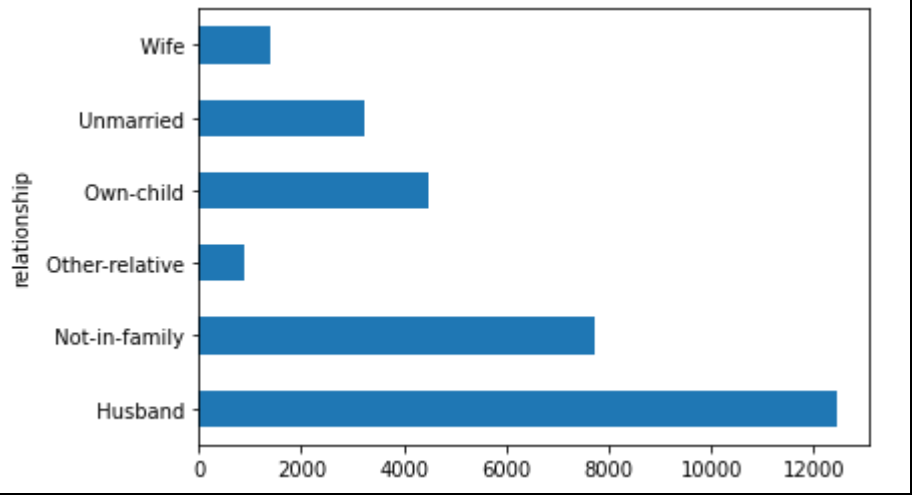
Figure2.1 First five rows of data

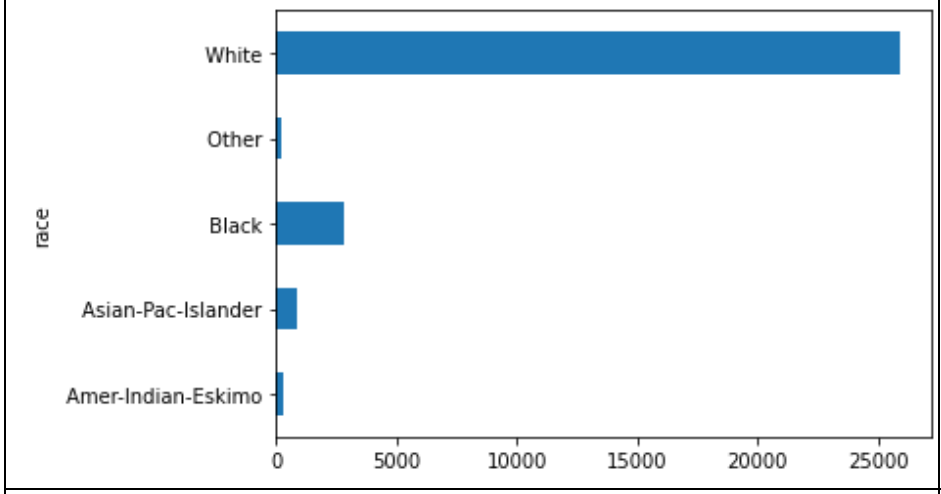
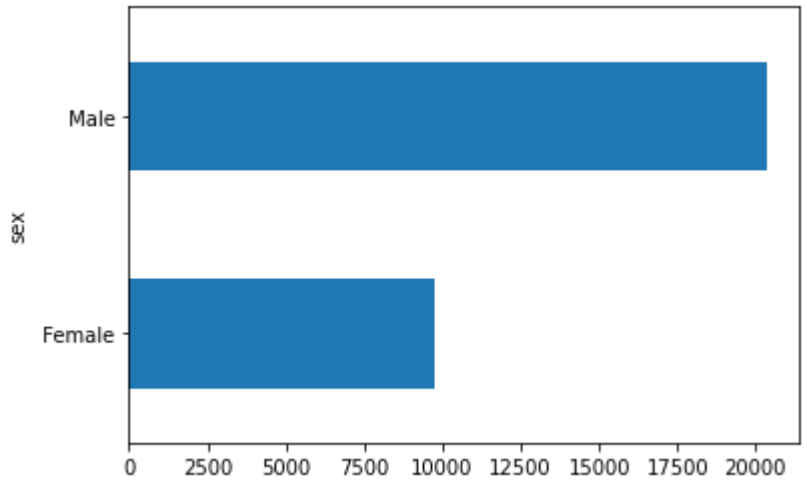
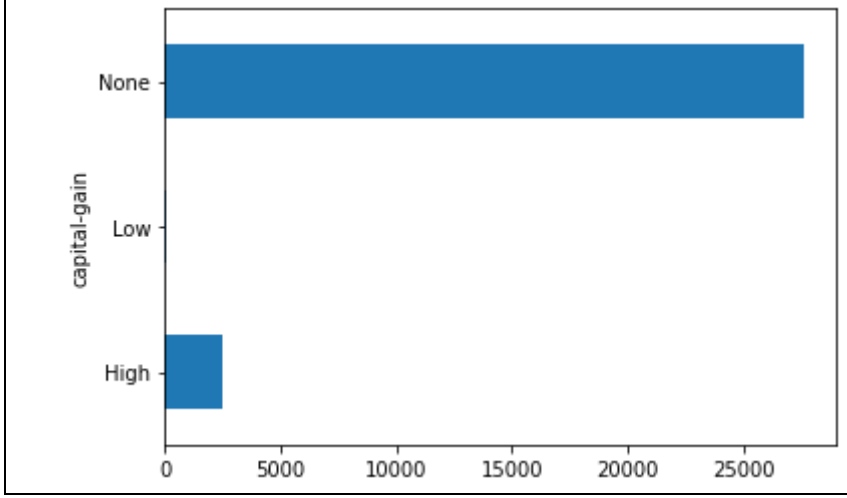
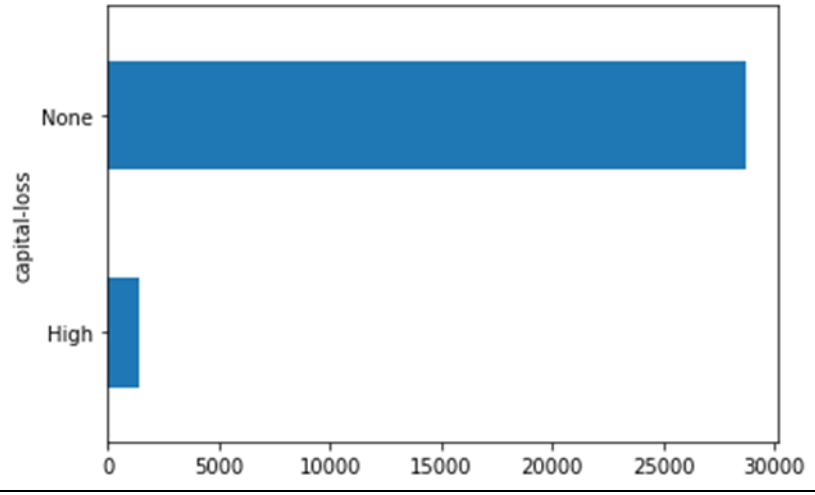
3 Data Distribution

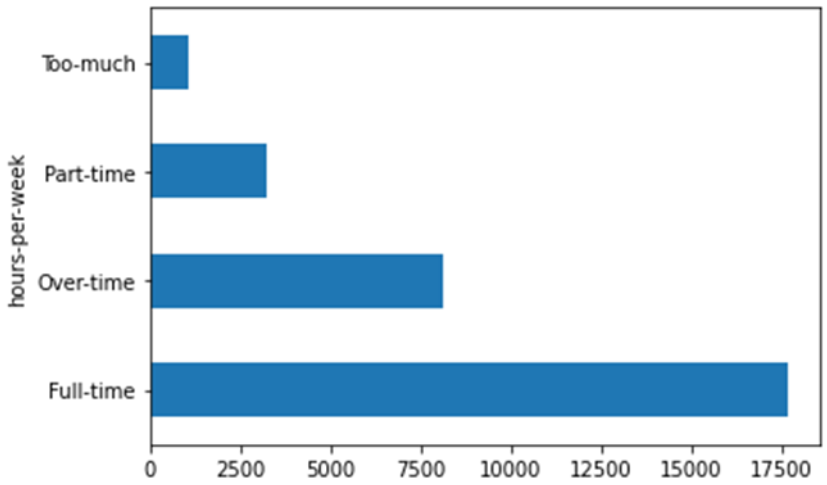
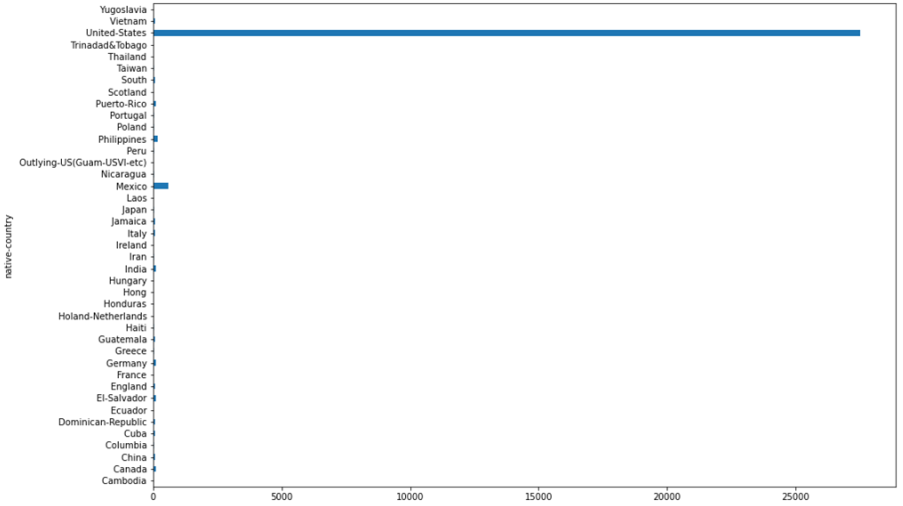
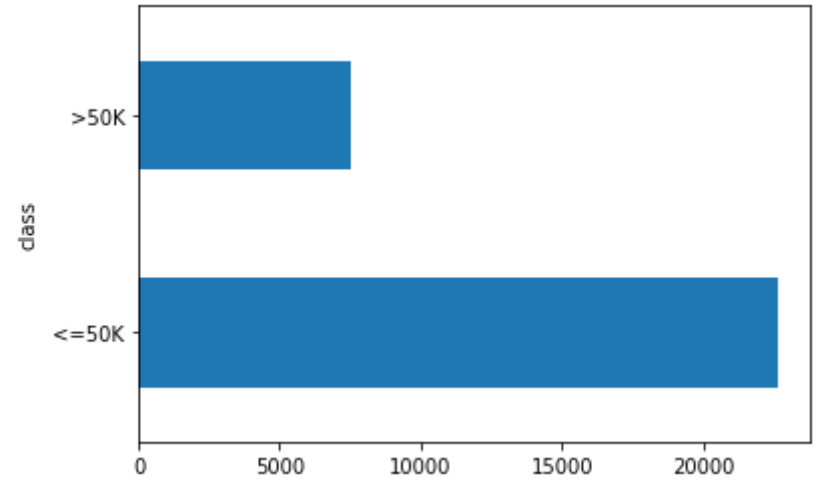
Table3.1 shows histograms of each column in the adult dataset. From the age histogram, we can see that the number of middle-aged (26-45) people is the largest, above the half of the total, followed by senior (46-65) and young people (1-25). Old people are the smallest group in this dataset. In the education histogram, it is clear that most people work for private companies. In the education histogram, nearly 10,000 people are high school graduates, followed by some college graduates and graduates with bachelor's degree. The histogram of marital status shows that about 14,000 people are married while around 10,000 people are never married. Also, there are over 4,000 people who get divorced. People's job varies widely in the picture of the occupation column. In the relationship histogram, more than 12,000 people are currently husband and the number of males is about twice as large as that of females in this dataset. Moreover, most people investigated are white people in the U.S.. In the histogram of the hours-per-week column, more than half people work full time and 7,500 people even work overtime per week. In the histogram of the class column, we can see that people whose annual gross incomes are less than or equal to 50k dollars are three times as large as people whose gross incomes are over 50k dollars in one year.

Table3.1 Data distribution in each column

Age	Workclass
	

Education	Marital-status
	
Occupation	**Relationship**
	

Race	Sex
	
Capital-gain	Capital-loss
	

Hours-per-week	Native-country
	
class	
	/

4 Frequent Pattern Mining

In this project, I use Apriori and FP growth algorithms to mine frequent patterns from the adult dataset. Notice that in the last section, I find that in the capital-gain and capital-loss histograms, above 90% of values are none. It is therefore obvious that these 'None' values are frequent itemsets and I do not take them into consideration while mining frequent patterns.

Table4.1 shows the frequent patterns and strong association rules with the minimum support count threshold (min_sup) of **0.6*30162** (minimum support threshold is 0.6) and the minimum confidence threshold (min_conf) of **0.7.**

Table4.1 Frequent patterns

K-itemset	Frequent itemset
1-itemset	{'United-States'}, {'Private'}, {'<=50K'}, {'White'}, {'Male'}
2-itemset	{'White', '<=50K'}, {'White', 'United-States'}, {'United-States', 'Male'}, {'United-States', 'Private'}, {'White', 'Private'}, {'United-States', '<=50K'}
Strong association rules	
{'United-States'} ⇒ {'Private'}, conf = 0.732 {'White'} ⇒ {'<=50K'}, conf = 0.736 {'White'} ⇒ {'Private'}, conf = 0.738 {'United-States'} ⇒ {'<=50K'}, conf = 0.746 {'<=50K'} ⇒ {'White'}, conf = 0.843 {'Private'} ⇒ {'White'}, conf = 0.859 {'United-States'} ⇒ {'White'}, conf = 0.881 {'Private'} ⇒ {'United-States'}, conf = 0.903 {'<=50K'} ⇒ {'United-States'}, conf = 0.905 {'Male'} ⇒ {'United-States'}, conf = 0.911 {'White'} ⇒ {'United-States'}, conf = 0.934	

From Table4.1, we can draw some insights that are consistent with what is found in the section of data distribution. Over 60% people living in the U.S. covered in this dataset gain 50,000 dollars or less in a year. Above 60% people are white people and their annual gross incomes are not more than 50,000 dollars. More than 60% people are living in the U.S and they also work for private companies.

For strong association rules, we can see that if people are white people, then the probability of their annual gross income being below 50,000 dollars is high with the value of 0.736. Also, if people are living in the U.S., it is highly likely that their annual gross incomes are less than or equal to 50,000 dollars with the probability of 0.746.

5 Algorithm Comparison

5.1 Improve the efficiency of Apriori

In this project, I use two methods to improve the efficiency of Apriori algorithm. One method is transaction reduction, that is, removing the transaction if it does not contain any frequent k-itemsets. Another optimization is sampling without replacement. For the sampling method, I lower the minimum support count threshold to **0.5*100** (minimum support threshold is 0.5) and the sample size is **100**. Notice

that this method trades off some degree of accuracy against efficiency.
Table5.1 shows the running time of the Apriori algorithm before and after improvement. We can see that the sampling method did improve the efficiency of Apriori algorithm a lot, cutting the running time in half. Moreover, the results in Table5.2 show that the sampling Apriori algorithm can well reflect frequent patterns as the unsampling algorithm does. But for the transaction reduction, it did not improve the efficiency of the Apriori algorithm in the adult dataset. It is possible that each time we reduce the transactions, we have to go through every remaining transaction and check if it contains any of frequent k-itemsets. It is time-consuming although this method does reduce the transactions for the next scanning.

Table5.1 Running times of the Apriori algorithm before and after improvement

Apriori	Before improvement	Transaction reduction	Sampling
Running time(s)	1.231591	7.588946	0.501252

Table5.2 Frequent patterns using sampling Apriori algorithm

K-itemset	Frequent itemset
1-itemset	{‘United-States’}, {‘Private’}, {‘<=50K’}, {‘White’}, {‘Male’}, {'Full-time'}
2-itemset	{'White', '<=50K’}, {'White', 'United-States'}, {'United-States', ‘Male’}, {'United-States', ‘Private’}, {‘White’, ‘Private’}, {'United-States', '<=50K'}, {'<=50K', 'Private'}, {'Male', '<=50K'}, {'Male', 'Private'}, {'Male', 'White'}
3-itemset	{'<=50K', 'United-States', 'Private'}, {'Private', '<=50K', 'White'}, {'United-States', 'Male', 'White'}, {'Private', 'United-States', 'White'}, {'<=50K', 'United-States', 'White'}

5.2 Apriori vs. FP Growth

It is known that FP growth runs faster than Apriori under most circumstances since FP growth requires limited times of scanning a database, while Apriori has to scan a database over and over again before it reaches the break point, and it generates an intractable amount of candidate itemsets during each scanning.
Table5.3 shows the running times of the Apriori and FP growth algorithm in this project. If we only want to get the frequent patterns or itemsets in the dataset, the FP growth algorithm is much more efficient than the Apriori algorithm. However, if we want to get the frequent patterns ae well as the strong association rules in this dataset, the efficiency of these two algorithms is close. This is because when generating the strong association rules from FP growth algorithm, the support count value is being computed and it takes time. Therefore, this method might need further improvement.

Table5.3 Running times of the Apriori and FP growth algorithm

Algorithms	Apriori without rules	FP growth without rules	Apriori with rules	FP growth with rules
Running time(s)	1.152461	0.514621	1.231591	1.279599