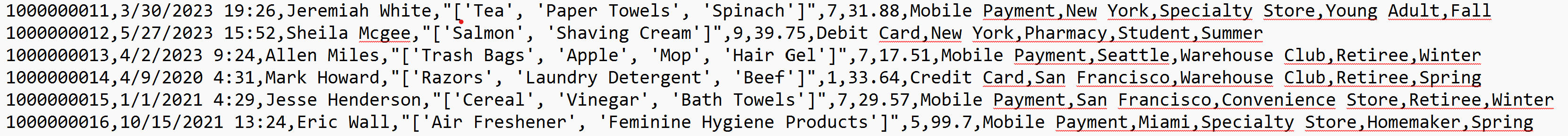
CS 528-02 Final Project Report Ashlin John

INTRODUCTION:

Market basket datasets often face privacy threats where hackers access two or more datasets from the same institution, and combine them to make inferences, which can further lead to revealing sensitive information of their customers. In order to prevent this, different privacy measures like anonymization and differential privacy are applied to the dataset to distort the data enough to make it useless to a third party individual. In this project, a retail transactions dataset of two-thousand transactions was acquired to perform safety measures on, while still maintaining the utility of the overall data. Further analysis is conducted on how well the privacy measures are implemented, such as adhering to k-anonymity and l-diversity requirements, and Laplacian noise addition efficiency. Once that is complete, association rule mining is done on the dataset to figure out if there are any patterns in the purchase that are implicating.   
  
CONTENTS:

Retail\_Transaction\_Dataset.txt is the input file from which all the transaction information is read from. The following is a snippet of how this .txt file looks like.



ReadFile.java is used to read the contents of the input file, while FileContent.java is used to identify the category of different information listed.   
  
Next, privacy measures are implemented in Anonymization.java and DifferentialPrivacy.java.

Since information like Customer Name and Payment Method are sensitive attributes, through anonymization (suppression), they were removed. There was no code specific for this segment because when the information was being read from the file, the sensitive attributes were ignored. Anonymization.java

Quasi-identifiers like Customer category and city were generalized. Date and time category was going to be generalized as seasons, but since seasons were a category already, Date and time were completely omitted. The following code snippet was used for this.

A screenshot of a computer code

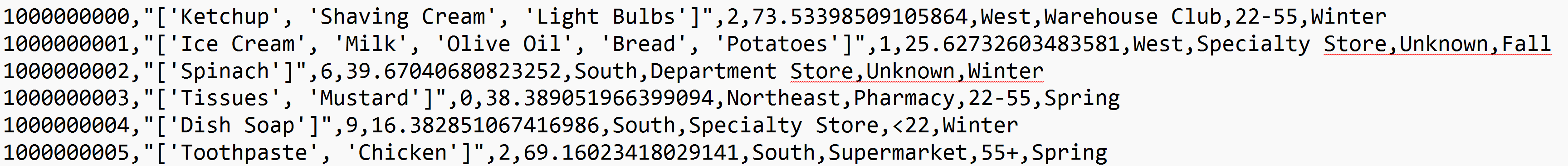
Description automatically generated

A screenshot of a computer code

Description automatically generated

For differential privacy, Laplace Mechanism was implemented. Noise was added to total items and total cost. The logic for Laplacian noise, and implementation are as shown below. Since the dataset was big, an Epsilon value of 1 was used for this.   
A screenshot of a computer code

Description automatically generated  
A java class called MainTest1.java was used to implement these measures. The altered and privacy implemented output (MainTest1\_Output.txt) was generated. Below is a snippet of its content.



In the output, all sensitive attributes are removed, quasi-identifiers are generalized, and noise is added to cost and item count. Below is a snippet on noise addition before and after.

A screenshot of a computer

Description automatically generated

A java class called SensitivityAndVarianceCalculator.java was created to calculate the sensitivity and variance od total items and total cost after noise was added. The following was the result.   
Sensitivity of Total Items: 19.0

Variance of Total Items: 9.360443750000012

Sensitivity of Total Costs: 99.68296234895148

Variance of Total Costs: 743.7130640830594

What this implies is that  
- A sensitivity of 19.0 suggests that adding or removing one transaction can change the total count of items purchased by up to 19 units.

- A variance of 9.36 means that the counts of total items purchased are relatively close to the mean value, with a moderate amount of variability around that mean.

- A sensitivity of 99.68 suggests that the total cost can change by up to 99.68 units when adding or removing one transaction.

- A variance of 743.71 means that the total costs across transactions are more widely spread out from the mean value, indicating a higher variability in total costs compared to total items.

In order to make sure that k-anonymity and l-diversity were intact, KAnonymityChecker.java and LDiversityChecker.java were implemented. In MainTest2\_Eval,java class, attributes of the checker classes were implemented on MainTest1\_Output.txt file’s content to check if it still meets the desired requirements. Once the process was done, an output file called Eval\_Output.txt displays if the requirements are satisfied or not. The logic for both checker classes is as below.

(MainTest1\_Output.txt is the file with privacy measures implemented.)

K-anonymity

A screen shot of a computer

Description automatically generated

l-diversity

A computer screen shot of text

Description automatically generated

Eval\_Output.txt content:

A black text with numbers

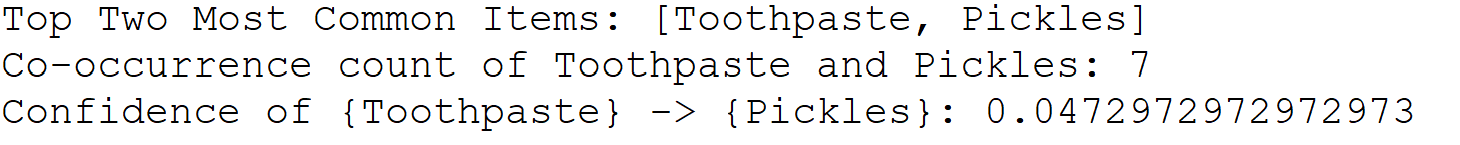
Description automatically generated with medium confidence

ExtractPurchasedItems.java was used to read just the items purchased, so any underlying patterns can be made. The output for this is recorded in PurchasedItems.txt.The contents of PurchasedItems.txt is as follows.

A white background with black text

Description automatically generated

This was done so association rule mining (AssociationRuleMining.java) could be implemented on the contents of the file to see if there is any underlying patterns. The only pattern that was there is as follows.



Files:

Retail\_Transaction\_Dataset.txt

MainTest1\_Output.txt

Eval\_Output.txt

PurchasedItems.txt

ReadFile.java

FileContent.java

Anonymization.java

DifferentialPrivacy.java

MainTest1.java

SensitivityAndVarianceCalculator.java

KAnonymityChecker.java

LDiversityChecker.java

MainTest2\_Eval,java

ExtractPurchasedItems.java

AssociationRuleMining.java

Website:

<https://www.kaggle.com/datasets/prasad22/retail-transactions-dataset/data>