A REPORT ON

DATA MINING ASSIGNMENT

DATA MINING

CS F415

SUBMITTED BY

NAME	ID
VIGNESH N.	2015A7PS0355P
DURJAI SETHI	2015A8PS0489P

Pre-processing

1. The process of pre-processing in this assignment entailed the deduction and separation of parameters of date, time(hh,mm,ss) for further clarification of data sets present in the given domain.
2. For each month the following steps were taken in the SPSS stream:

- First the data was sorted by **selling date** and if that is same then sort by **transaction ID**, this was done to visualize the data as a set of bills as all transactions of a bill will appear together.
- Then 3 new features were created i.e. extracting the **student segment** from ID number and extracting **hour number** and **day** from selling date.
- The data was analysed by using the data audit node and some crude observations which were made were:

    a. Most items bought had selling price $<$ **40**

    b. Most number of students came to eat after **10:00 pm**

    c. In almost all the months, students of segment **F2** and **F3** were the most frequent visitors of ANC.

- The statistics node was also used to try to get **pearson correlations** between attributes to see if any pair of attributes were correlated or not.

Problem-1

For all problems, the stream is in Question 1 folder.
For this particular problem, which demands the need to incorporate the concept of dynamic pricing, the following steps were adopted, taking into consideration the given constraints:

Approaching this problem also meant that a perfect trade-off between minimization of penalty and maximization of revenue needed to be ensured. For the same, we used certain techniques from the following domain:

- Except columns Item ID, Quantity, Degree, Hour number, rest all were filtered out. The table generated by this filtering(**monthname_selected_features.csv**) was then processed by python script **for_problem1.py**. This script basically repeats a row in the table passed to it quantity number of times i.e. if the quantity is 2 in a row then repeat that row 2 times in the output csv file(**aug_selected_features_modified.csv**). This was done for the application of apriori algorithm to the data.

- Then the quantity field was filtered out in the spss stream. A **dummy field** (all row values for the field is set to true) was added using derive node. The reason for adding this is explained below.

- Apriori algorithm was applied on this table. Each transaction is a 3 itemset (item id, student degree, hour number). Since we wanted to find the frequent 3 itemsets in this table and not a rule, for this reason a dummy field was added with all row's values set to true. The set to flag node is used to convert the table to a presence absence matrix.

- Minimum antecedent support for the finding frequent itemsets was set to **0.4%** since at this minsup, we were getting adequate amount of 3 itemsets (item id, degree, hour number) for which we can change the price.

- After this, we manually increased the prices of items by **10%** at the hours and for the student segments which were frequent 3 itemsets in the months of August to November. The script **calculate_profit.py** then takes as input the **decSales.csv** file and also opens the **newPrices.csv** file. The script calculates and prints the **penalty** and **percentage** increase in revenue.

Problem-2

For each month the following steps were taken in the SPSS stream:

1. Except the columns, Bill No, ItemID, final rating, day all other columns were filtered out. The resulting table's rows were first grouped by day and then by bill number and we concatenated the item ID values (**using** '_') of a single bill. We did this to convert the data into a transaction data where each row is a bill there is one column containing all items bought under that bill. Only that column containing the item IDs of a bill was kept and rest of the columns discarded.
2. Now this transaction data table was processed by python script **convert_to_presence_absence.py** which converts the transaction data to presence absence matrix.
3. The presence absence matrix is read and then apriori algorithm is applied on that. The support is **0.1%** and the rule confidence is **75%**. This was done because we wanted to get those combination which have **high confidence** and **low support** (i.e. the item combinations which occur rarely but when the occur, they occur together). Some combinations were found using this approach.
4. After finding the combinations, the combinations were put in csv file **combos.csv** and this file was processed by the python script **loss_in_profit.py**. This file first processes the **combos.csv** file to put all combos in a list. Then the combo prices are calculated and the output csv file **problem2.csv** is created which has the list of combos along with their prices. Then the **percentage loss in revenue** is calculated and is output on stdout.
5. Another way to find combinations is to find the **average rating** of each item across all months and then divide the items into low rating ones and high rating ones.
   a.Except item ID and final rating, all other attributes were filtered out.
   b.Then tables of all four months were appended together and sorted on the basis of item ID. Then the resulting table was grouped by item ID field and in each group, the mean final rating was calculated. So, the resulting table has each row as an item ID and its mean final rating.
   c.This table was divided into two tables, one contains items of low rating and the other table contains high rating items. The threshold for low rating items is <= 2.8 mean rating. First we set the threshold to 2 but then only 5-6 items were coming in the low rating table.

Problem-3

- To see the trends in the data provided, graphs were drawn to see the visual representation of the data. Graph was drawn between selling price and final rating which showed that most of the items bought fall in the selling price range of 0 to 50.

- Another observation which was made was by plotting graph between selling price and quantity. The graph showed 2 different clusters, one in the selling price range of 0 to 40 and quantity range of 0 to 200(it is the overall quantity of an item bought in a month), the other cluster is in the selling price range of 80-100 and quantity of 0-100. This graph can be used to interpret which student segment prefers food items in which range.

- Another analysis we did was to find the contribution of each item to the total revenue in a month. This analysis also helped in solving problem 1, since the results in problem 1 corresponded to those items which contribute more to the total revenue.

- The record of cash payments when quantified, showed us the results that there were high number of such transitions in the month of August, which was because the ID cards take a couple of weeks to be distributed. The month of September sees an increase of payments because of the presence of fests like Junoon and BOSM. There is a significant decline of such transactions during October. Still the number remains high, due to transactions during the fest: Oasis. November sees a plummeting decline, where the cash payments fall to more than half the October sales. This can be credited to the busy schedule of the students owing to more evaluatives and end semester preparations and no fests. December further sees transactions falling to half, due to the period of comprehensive examinations going on.