Data Challenge - Detailed Report

CS F415 - Data Mining

Abhay Koushik --- 2015A7PS0056P

Gautham V --- 2014B4A70637P

Problem 1:

1. **Preprocessing** : The given data was not nice, so it was processed. Please refer `prep1.py` and `alpha_ranges.py` .
    - Split student ID to obtain category
    - Split the DateTime value to obtain Date and Time separately
    - Split Time to get hours alone (because 16:00:00 to 16:59:59)
    - Changed hours (00->24, 01->25, 02->26) to get ordinal hour values *Note: this takes a long time, rowwise operation*
    - Added prices to `decSales.csv` so that we can obtain base revenue in December
    - Used prices in `decSales.csv` to find the maximum price increase possible (*10% or Rs. 10, whichever is lower*)
    - Combined months data into Training and Test data.
    - Combined months to `full_min.csv` which has only ItemID, Student_Type, and Hour.
2. **Data Mining**: Refer `Stream1.str` and `DM1.R` .
    - `Stream1.str` :
        - Used Decision Tree Classifiers `C5.0 Node` and noted down the rules obtained.
        - Looked at distribution of ItemIDs and Student_Types.
        - Used `Decision List Node` to obtain rules for most popular ItemIDs
        - But ultimately, this didn't lead to any good profits, so this was eschewed.
    - `DM1.R` :
        - Used the apriori method from the arules package.
        - Using base parameters in the *apriori* `(support = 1e-5, maxlen = 5)` and *eclat* methods `(support = 1e-5, minlen = 3)`
        - Obtained association rules and frequent itemsets.
        - Used these rules (along with their support and count) to form the

pricing scheme.

3. **Post-Processing**: Refer `changehrs.py`, `penalty.py`, `changeprices.py`.
    - `arules1.csv` is obtained from the AR Mining, it is processed to get `preArules.csv`.
    - Now `preArules.csv` has a lot of rules, but each rule has low support and spans only an hour.
    - The top *n* rules are obtained from `preArules.csv`.
    - These rules are then combined (so that they span multiple hours) and the rules which are "less profitable" are removed.
        - `supertop100rules.csv` -> top 100 rules from `preArules.csv` have been combined and the better ones have been taken.
        - `regtop100rules.csv` -> top 100 rules from `preArules.csv` have been combined and all have been taken.
        - `reg` rules have far higher penalty than `super` rules, for the same value of *n*.
        - The values *FP* and *FP2* are meant to show if a rule is "good" (low penalty, good support, good profit) or not.
    - The `super` and `reg` rules generated are then mixed with `PREdecSales.csv` to obtain the new revenue and penalty.
        - SciPy's `optimize` was used to attempt and find optimum weights to minimize penalty and maximize profit.
        - But prices were raised to the maximum for each item in its time slot, so penalty is very high in each case.
    - Results were stored in `RuleData.csv` and the rule with the best "Niceness Coefficient" was chosen.
    - `newPrices.csv` was accordingly edited with the updated prices from the best rule. **Note:** `newPrices.csv` does not have any provision for choosing the target student segment eg. F2.