OR 560 – Project Report

Using customer sentiment to predict customer satisfaction

By

Addagarla Suraj

Bendale Priyanka

Narayanan Satyajit

Sinha Mallika

1. Executive Summary

Through this project an attempt has been made to help Lenovo predict the customer satisfaction for their products with the help of customer sentiment data. Net Promoter Score (NPS) is a key indicator of a product or company's performance in the market. Lenovo quantifies customer satisfaction for a product in terms of product NPS (pNPS). The evolution of the sentiment and its relationship with customer satisfaction were studied and its key drivers were identified.

A Markov decision model was developed to look at the evolution of sentiment over time and to suggest any intervention that would be beneficial. The five components of the MDP (State Space, Action space, Epoch, Transition probabilities and Reward space) were defined as per standard assumptions and an optimal policy of 'Do Nothing' was derived using Policy Iteration. As some of the parameters were assumed, they were varied to check the validity of the result and it was found that the optimal policy for each state remained the same.

Key variables such as frequency of a sentiment occurring for each taxonomy and the evolution of sentiment were used to predict the pNPS. A model was built using Principal Component Regression (PCR) method to predict the pNPS each for Consumer and Commercial products. Upon validation, the model gave an accuracy of 5.34% and 8% for Consumer and Commercial models respectively. The equations obtained from these models were used to calculate the pNPS values for the 4 test products.

Stochastic and predictive models were created which captured the effects of evolution of sentiment over time and the variation in taxonomies to estimate customer satisfaction. The equations obtained from the model can be used to predict the pNPS of any product within the Commercial and Consumer groups. Furthermore, areas of strength and improvements among taxonomy levels were identified through a word cloud.

2. Introduction:

NPS, measures customer experience and predicts business growth. It provides companies with a simple and straightforward metric that can be shared with their front-line employees. Customers are surveyed on one single question. They are asked to rate on a 0-10 scale the likelihood of recommending the company or brand to a friend or colleague. Based on their rating, the respondents are grouped as follows:

- **Promoters** (score 9-10): They love the company's products and services. They are the repeat buyers, are the enthusiastic evangelist who recommends the company products and services to other potential buyers.
- **Passives** (score 7-8): They are somewhat satisfied but could easily switch to a competitor's offering if given the opportunity. They probably wouldn't spread any negative word-of-mouth but are not enthusiastic enough about your products or services to actually promote them.
- **Detractors** (score 0-6): They are not particularly thrilled by the product or the service. They, with all likelihood, won't purchase again from the company, could potentially damage the company's reputation through negative word of mouth.

Subtracting the percentage of Detractors from the percentage of Promoters yields the Net Promoter Score, which can range from a low of -100 (if every customer is a Detractor) to a high of 100 (if every customer is a Promoter).

The reason that any company (including Lenovo) wants to maintain a record of its customer satisfaction levels is that it would cost a company much more money to win back a detractor as opposed to simply keeping promoters on board. Having an NPS score would give an idea of where the product, and the company in general, stands among the public. Higher Net Promoter Scores tend to indicate a healthier business, while lower Net Promoter Scores can be an early warning to dig deeper into potential customer satisfaction and loyalty issues.

Poor NPS could lay emphasis on rectifying organizational loopholes that would benefit the company on the whole. The power of the NPS lies in its simplicity but unless the data is properly analyzed, it is difficult to trace out the factors that turned simple customers into promoters or detractors.

The following key information was provided by Lenovo to help to identify the relationship between customer sentiment and NPS:

- Sentiment Data: Information collected from web scrapes and third-party apps which had data regarding the following:
  - Comment ID
  - Sentiments that the software recognized and allotted to various product features in that comment
  - Levels of classification of the product features
  - Product name
  - Date (MM/DD/YYYY)
  - Star Rating assigned by the customer
- Survey Data: Official survey responses to many questions important among which are:
  - Survey ID & date (MM/DD/YYYY)
  - Product & Brand NPS
  - Product name & Series
  - Additional questions like satisfaction levels with various features, service aspects, purchase process, etc.

3. Methodology

3.1 Markov Decision Process

The objective of formulating a Markov Decision Model was to understand the evolution of customer sentiment and the need to take appropriate action. Given the inherent uncertainties in the existing process, a stochastic model would accurately capture the development of sentiment. Markov Decision Process is a mathematical framework for modeling decisions where outcomes are probabilistic rather than deterministic but can be influenced by the decision maker. It also suggests the best actions to be taken based on the utilities available. The components of the present Markov Chain are defined below:

- ***State space (S)*****:** This includes the key variables of interest at any given point in time. Since the objective is to assess the evolution of sentiment over time, the state space comprises of the three customer sentiments i.e. Positive, Neutral and Negative.

- ***Action space (A)*:** This contains the set of possible actions that Lenovo could take when in each state. The two possible actions include ‘Do Nothing’ and ‘Intervene’. The future state (i.e. sentiment at a later stage) is impacted by the decision taken in the present state.
- ***Decision epoch (T)**:* This defines the time intervals over which the data is gathered and analyzed, and when a potential action can be initiated. The MDP in discussion is modelled as a discrete-time Markov chain by taking a month as an epoch. Since the data available is spread over sixteen months, aggregating at a month level was justified.
- ***Transition probability (P)**:* It represents the probability of moving from one state to another under a given action. This matrix is computed from the transition of sentiment observed in the data provided. The transition matrix for both actions has been provided in the Appendix A.
- ***Rewards (R)***: This parameter accounts for the expected cost of taking an action in a state. The rewards (or costs) have been considered for each State-Action combination considering both the customer satisfaction and costs incurred by Lenovo. Detailed description has been provided in Appendix B.

Assumptions:

- The sentiment data available corresponds to ‘Do Nothing’ scenario
- For any given state, ‘Intervene’ action improves sentiment by a greater magnitude than the corresponding ‘Do Nothing’ scenario
- Any intervention action by Lenovo invokes the same cost irrespective of the state the system is in
- It is more difficult to convert negative sentiment into positive sentiment than converting negative sentiment to neutral sentiment. This is based on the presumption that it is more difficult to win over a detractor than to maintain promoters on board

Optimal Policy: To define the best action to be taken in each state, *Policy Iteration algorithm* was employed (with a discount factor of 0.9) which solves for an optimal policy by considering an infinite horizon decision problem. Optimal policy came out to be ‘Do Nothing’ for all the three states (for both the set of products). Appendix C can be referenced for further details.

3.2 Variable Selection

The Markov decision model helped estimate the optimal policy, but the value of pNPS still could not be predicted. As a next step, potential predictors of pNPS were identified.

3.2.1 Taxonomies as predictors:

To understand how customer sentiments, affect the pNPS, it was decided that the frequency of sentiments be used as predictors of pNPS. In order to capture the variance between taxonomies, the frequency of sentiments was broken down at a taxonomy level to be used for further analysis. Because of the high multicollinearity between taxonomies, grouping of taxonomies based on correlation and class of taxonomy was done. Examples in Appendix G.

3.2.3 Evolution of Sentiment as a predictor: *(Why, how, Examples)*

To understand the impact of development of sentiment over time on the pNPS, an Evolution of Sentiment (EOS) variable was created. The transition probability matrix for each product series was computed in a manner similar to P-matrix of the MDP model. Then the sentiment distribution for each series in the first month (q) was computed from the data available and the distribution in the following months was obtained using the Chapman-Kolmolgorov rule ($q^n = q*P^n$ where q^n = sentiment distribution in n^{th} epoch). Examples in Appendix G.

To calculate EOS, the distribution of sentiment in the first month should be known. The sentiment distributions across various months are then used as independent variables in the regression analysis.

3.2.4 Calculation of pNPS as a response variable

The pNPS value for a month is calculated as cumulative subtraction of promoter % and detractor % till that month. It is assumed that the sentiments will reflect on the pNPS after a lag of five months. Also, Five-month lag provides maximum overlap between sentiment and survey data.

After deciding the dependent and independent variables, the level of data to be used for prediction needed to be decided. Data at a Series-Month level was considered. Independent and dependent variables were calculated at this level each for Consumer and Commercial data.

3.3 Prediction

3.3.1 Multi Linear Regression

MLR was implemented using the previously decided predictors to predict pNPS, however the model failed due to multicollinearity and sub optimal solutions. In case of multi linear regression for fewer number of variables, the collinearity between each variable is checked. If the variables have collinearity, they are eliminated from the final dataset. However, in case a large number of variables exist, checking collinearity between each variable is not feasible.

3.3.2 Principal Component Regression (PCR)

Principal Component Analysis (PCA)

In order to eliminate issues like multicollinearity and sub optimal solutions, principle component analysis (PCA) was used to figure out values of set of linearly independent uncorrelated variables called principle components. PCA eliminated these two issues due to the orthogonal transformation property of the principle components. The components were then used as regressors to perform principle component regression. In the present context, each product type (Consumer and Commercial) had multiple taxonomy levels.

Using components to perform regression

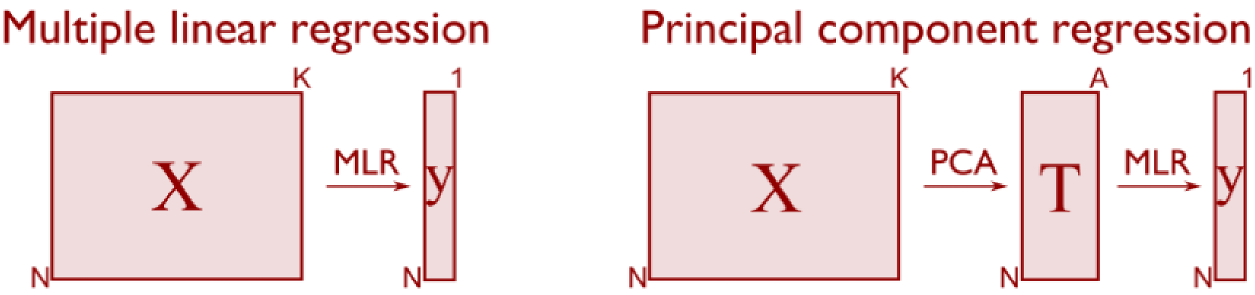


Figure 1: MLR vs PCR

Using PCA for regression is a 3-step process. First, PCA is performed on the original data of independent variables, represented by table X in *Figure 1*, to obtain principal components, each being a linear combination of all the *K* variables used in the analysis. PCA was performed on Python for 110 variables for Commercial and 138 independent variables for Consumer data.

Of the components obtained, the most significant ones (*A* components as in *Figure 1*) were chosen based on the proportion of variance they explain and their relationship with the dependent variable (*Y*). This was done by calculating the cumulative eigenvalues which denote variance in the data explained by a particular component. The *Scree plot* shows the spread of cumulative eigen values.

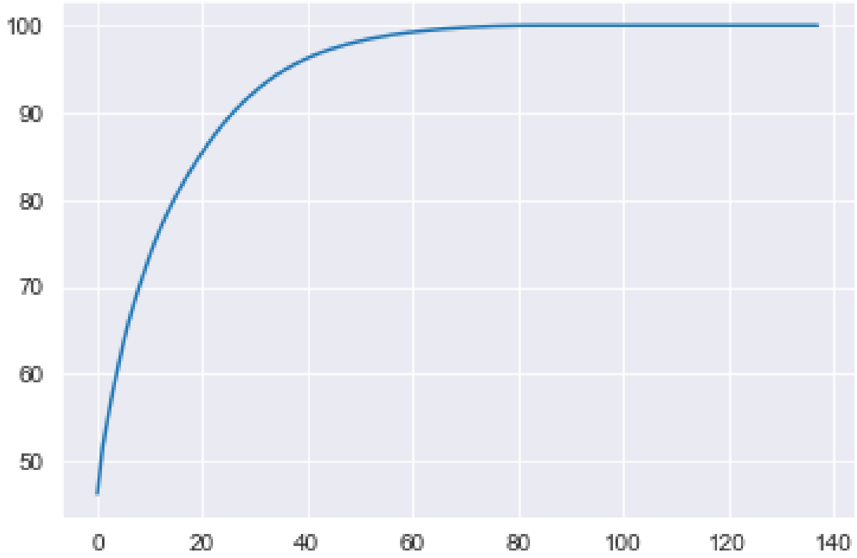


Figure 2: Scree Plot for Consumer data

Figure 2 shows that the first 50 components explained ~97% of variance in data. The other principal components corresponding to small eigenvalues were omitted. The same procedure was followed for Commercial data too. The set of reduced components forms dataset *T* (as shown in *Figure 1)* which is then used for regression.

4. Results:

4.1 Prediction equation and accuracy:

After obtaining the principal components for Commercial and Consumer data, the columns in *T* from PCA were used as the data source for the usual multiple linear regression model. The dataset was divided into training and testing data in order to validate the robustness and accuracy of the model. The ratio of split taken was 80/20 for training and test respectively.

For the 50 components of Consumer data, multiple iterations of regression were performed to observe Probability significance value (P-value) and Adjusted R-square values. These were used to select the 31 significant components which contributed to ~86% of variance. An equation was formed using the coefficient values obtained from Ordinary least squares (OLS) regression result. The final model obtained had an Adjusted R-square of 60.7% and a Root Mean Square Error (RMSE) of 5.34% while predicting for the validation dataset. Similarly, 22 significant components were obtained for Commercial data which lead to an Adjusted R-square of value of 60% and an error of 8%.

Table 1: Summary of PCR results

Information	Commercial	Consumer
# of components used to predict pNPS	22	31
Adjusted R-squared	0.599	0.772
Validation accuracy (RMSE)	8.001	5.34

The complete equations for prediction of pNPS have been mentioned in Appendix D. Below is the condensed version of the equations obtained for prediction of pNPS for Consumer and Commercial:

$\text{pNPS}_{\text{Consumer}} = 30.685 + (0.385)*C1 + (-1.379)*C2 +\ldots..+ (-1.33)*C30 + (-2.02)*C31$

$\text{pNPS}_{\text{Commercial}} = 42.202 + -0.778C1 + 0.563C2 +\ldots..+ (5.93)*C21 + (0.92)*C22$

Using these equations, predicted pNPS values were calculated for the required products (in Table 2).

Table 2: Predicted pNPS values for test products

Product Type	Product	Month of prediction	Predicted pNPS
Commercial	T480	Mar-18	60.100
		Apr-18	59.920
		May-18	54.533
	X1 CARBON 2018	Mar-18	*80.983*
		Apr-18	21.681
		May-18	36.506
Consumer	IDEAPAD 120S 11	May-18	27.312
		Jun-18	27.029
		Jul-18	26.803
	YOGA 920	May-18	*33.571*
		Jun-18	25.237
		Jul-18	34.280

4.2 Markovian model results

Since the rewards and transition matrices were assumed, the policy of 'Do Nothing' was arrived at. Once Lenovo implements some corrective or sentiment improvement actions (if not in place already), it can then take into account the actual flow of sentiment (P-matrices) and the costs incurred using which standard and more responsive policies can be developed. The transition probability matrix that was computed represents the evolution of sentiment over time which was used as a key variable in the PCR.

5. Recommendations:

- One-to-one correspondence between sentiment respondent and survey respondent is recommended. This will help track the perception of that particular respondent. Lenovo could allocate a unique number to the email-id of each customer. When a customer provides feedback on any source using his or her email-id, it can be backtracked using this unique number. This will help in tracking the perception and in turn the sentiment of the respondents

- More weight can be given to star ratings rather than simply looking at the software-generated sentiments as it reflects the true opinion of the customer
- Survey questions can be improved/updated in response to the issues being faced by customers i.e. they can be dynamic in nature rather than passing out the same surveys throughout the product life cycle
- The algorithm for sentiment analysis can be improved as it incorrectly classifies some sentiments (in case of sarcasm for example)
- Areas of strength and improvements among taxonomy levels were identified through a word cloud. *Battery* and *Keyboard* over indexed on the negative sentiment and must be looked into (Refer Appendix H)

6. Conclusions:

Stochastic and predictive models were created which captured the effects of evolution of sentiment over time and the variation in taxonomies to estimate customer satisfaction. The equations obtained from the model can be used to predict the pNPS of any product within the Commercial and Consumer groups. However, regression analysis has no consistency check on the new observation's x-values. It simply calculates a direct prediction for y-new, no matter what the values are in x-new. A definite Markov Chain could be developed if the model parameters are known (actions and their rewards).

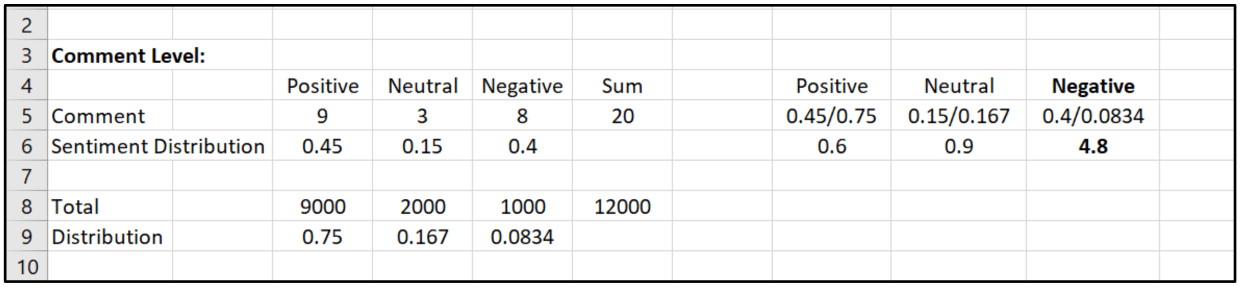
Appendix A: Transition Probability Matrices

The transition probability of moving from state A to state B is defined as follows:

$$P(State\ A \rightarrow\ B) = \frac{\#\ of\ transitions\ from\ A \rightarrow B}{total\ \#\ of\ transitions\ from\ state\ A}$$

There exists a probability matrix corresponding to each action.

‘Do Nothing’: A single transition matrix was computed for all products in each class i.e. one each for consumer and commercial products. The sentiment dataset comprised of sixteen months (05/2017 - 08/2018) worth of data and almost every comment had several sentiments attached to it, each of which was associated with a particular product feature mentioned in that comment. The percentage of each sentiment across every comment was calculated first (say r1) and then across all products together (say r2). The sentiment with the highest r1-to-r2 ratio across each comment was allotted as the ‘net sentiment’ for that comment.



2								
3	**Comment Level:**							
4		Positive	Neutral	Negative	Sum	Positive	Neutral	**Negative**
5	Comment	9	3	8	20	0.45/0.75	0.15/0.167	0.4/0.0834
6	Sentiment Distribution	0.45	0.15	0.4		0.6	0.9	**4.8**
7								
8	Total	9000	2000	1000	12000			
9	Distribution	0.75	0.167	0.0834				
10								

Figure A1: Sentiment aggregation across comments

For example, in the figure shown, even though the comment seems to have a positive sentiment (going by the maximum percentage). However, when compared to the overall distribution, this comment is predominantly negative (since $(r1/r2)_{negative} > (r1/r2)_{positive}$). Later, a month-wise distribution of sentiments was obtained using which a ‘net sentiment’ was assigned to each month in the same manner as before. From this monthly sentiment data, the probability transition matrix was obtained by considering a lag of one month (epoch).

Table A1: Transition probability matrix for Commercial products

P (Do Nothing)	**Positive**	**Neutral**	**Negative**
Positive	0.25	0.25	0.5
Neutral	0.4	0.2	0.4
Negative	0.167	0.334	0.5

Table A2: Transition probability matrix for Consumer products

P (Do Nothing)	**Positive**	**Neutral**	**Negative**
Positive	0.25	0.25	0.5
Neutral	0.142	0.714	0.142
Negative	0.25	0.5	0.25

‘Intervene’: Since prior data wasn’t available corresponding to any action taken, slight changes were made to the ‘Do Nothing’ P-matrix by decreasing the probability of any sentiment moving towards negative sentiment, and this decrease is then adjusted into the probabilities of going into positive and neutral sentiments. Also, it was assumed that the proportion of negative sentiment shifted to positive sentiment is smaller than the proportion being shifted to the neutral sentiment (in line with our assumptions). ‘a’ and ‘b’ percent of negative sentiment was adjusted into positive and neutral sentiments respectively. The following table shows the transition matrix for ‘Intervene’ action where the term x_{ij} represents the transition probability from state ‘i’ to state ‘j’, taken from the corresponding ‘Do Nothing’ matrix. Note that the probability of going into negative sentiment state is reduced by (a + b)%.

Table A3: Transition probability matrix for 'Intervene' action

P (Intervene)	**Positive**	**Neutral**	**Negative**
Positive	$x_{11} + a*x_{13}$	$x_{12} + b*x_{13}$	$x_{13}(1 – a – b)$
Neutral	$x_{21} + a*x_{23}$	$x_{22} + b*x_{23}$	$x_{23}(1 – a – b)$
Negative	$x_{31} + a*x_{33}$	$x_{32} + b*x_{33}$	$x_{33}(1 – a – b)$

Table A4: Example of P(Intervene) for (a = 5%, b = 10%)

P (Intervene for a = 5%, b = 10%)	**Positive**	**Neutral**	**Negative**
Positive	0.275	0.3	0.425
Neutral	0.42	0.24	0.34
Negative	0.192	0.384	0.425

As is evident from the example, the probability of any sentiment going into negative sentiment is reduced (by 15%) and the probabilities of going into the other two sentiment states increase. If in case Lenovo can achieve such a probability transition matrix by intervening, it'll be worth saying that intervention is beneficial.

Appendix B: Rewards

Rewards were assigned based on the current state ‘i’, and the action ‘a’ taken given that the system was in state ‘i’. This gave rise to six reward combinations resulting from the three states and two actions. These rewards were divided into two categories - one to account for the change in customer sentiment (R_{CS}) and the other reflected the monetary costs that Lenovo could incur (R_{MC}). The reward space remains the same for both consumer and commercial products.

Under the customer satisfaction category, the six combinations were ranked according to the relative improvement in customer sentiment (i.e. moving away from negative sentiment). Also, ‘Intervene’ action is assumed to improve or at least maintain the sentiment (i.e. the probability of sentiment going towards negative is minimal or reduced). In the monetary costs’ category, the combinations having ‘Intervene’ action were assigned a higher ranking than ‘Do Nothing’ cases since intervention by Lenovo usually involves some service action that incurs costs.

Once the ranks were assigned, numerical weights were derived from the ranks using the rank sum method. If ‘n’ is the total number of ranks, a combination having a rank ‘r’ receives the weight (n – r + 1). These weights were then converted into rewards by normalizing onto a 0-10 reward scale. The final reward corresponding to each combination was obtained by subtracting the monetary costs from satisfaction rewards i.e. $R_{Total} = R_{CS} - R_{MC}$ (the monetary rewards were taken negative since they’re an investment by Lenovo).

Table B1: Rewards for State-Action combinations

State (S)	**Action (A)**	**Customer Satisfaction (CS)**			**Monetary Costs (MC)**			**Net Reward R (S, A)**
		Rank	**Weight**	**R_{CS}**	**Rank**	**Weight**	**R_{MC}**	
Positive	**Intervene**	1	4	10	1	2	(10)	0
Neutral	**Intervene**	2	3	6.67	1	2	(10)	(3.34)
Positive	**Do Nothing**	2	3	6.67	2	1	0	6.67
Negative	**Intervene**	3	2	3.34	1	2	(10)	(6.67)
Neutral	**Do Nothing**	3	2	3.34	2	1	0	3.34
Negative	**Do Nothing**	4	1	0	2	1	0	0

Appendix C: Optimal Policy

In order to find an optimal policy for each state, policy iteration (PI) algorithm was employed. PI takes in a random initial policy, the rewards as per that policy and computes an optimal policy according to the maximum expected utility principle. Then it iteratively performs two steps: value determination, which calculates the utility of each state given the current policy, and policy improvement, which updates the current policy if any improvement is possible. Each of these equations is mentioned below.

Value Determination: $V_{\delta}(S_i) \ = \ R(S_i, A_{\delta,i}) \ + \ \beta * \sum_{j=1}^{N} P(S_j \mid S_i, A_{\delta,i}) * V_{\delta}(j)$

Policy Improvement: $T_{\delta}(S_i) \ = \ \max\{R(S_i, A_i) \ + \ \beta * \sum_{j=1}^{N} P(S_j \mid S_i, A_i) * V_{\delta}(j)\}$ where

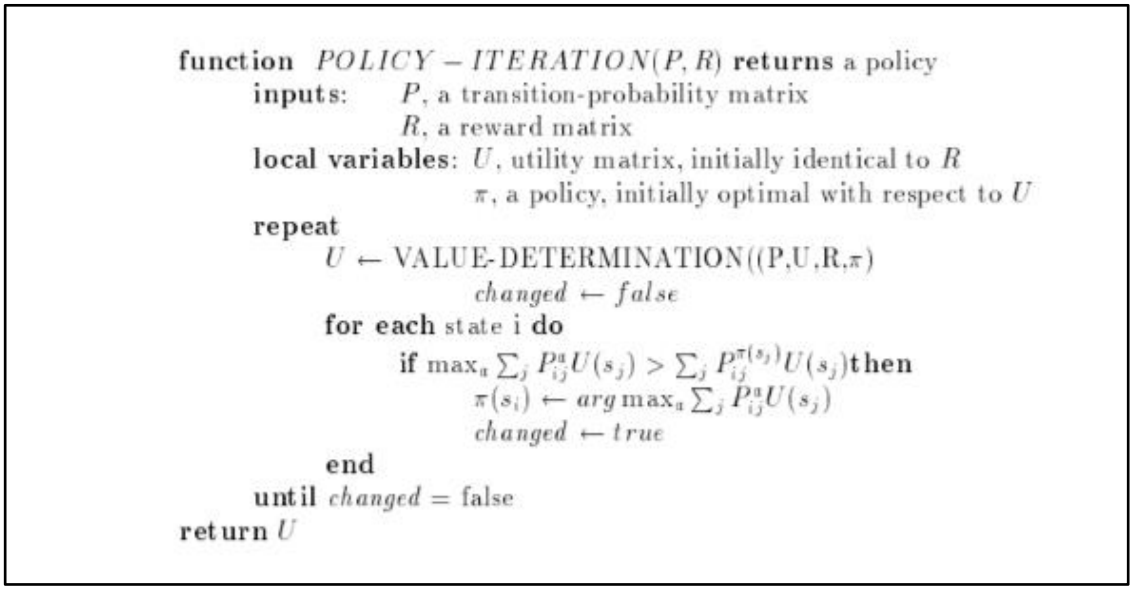
A_{δ} : Action space corresponding to optimal policy 'δ'

V_{δ} : Value determination as per optimal policy 'δ'

T_{δ} : Policy improvement as per optimal policy 'δ'

$P(S_j \mid S_i, A_{\delta,i})$: Probability of going to state S_j from state S_i given that action $A_{\delta,i}$ was taken as per optimal policy 'δ'

β : Discount factor employed in Policy Iteration (0.9)



```
function  $POLICY - ITERATION(P, R)$ returns a policy
    inputs:      $P$, a transition-probability matrix
                 $R$, a reward matrix
    local variables: $U$, utility matrix, initially identical to $R$
                     $\pi$, a policy, initially optimal with respect to $U$
    repeat
        $U \leftarrow$ VALUE-DETERMINATION((P,U,R,$\pi$)
                     $changed \leftarrow false$
        for each state i do
            if $\max_a \sum_j P^a_{ij} U(s_j) > \sum_j P^{\pi(s_i)}_{ij} U(s_j)$then
                $\pi(s_i) \leftarrow arg \max_a \sum_j P^a_{ij} U(s_j)$
                $changed \leftarrow true$
        end
    until $changed =$ false
return $U$
```

Figure C1: Policy Iteration Algorithm (taken for β = 1) [Ref. 6]

The value determination equations Vδ(s) for all states are first calculated by taking an initial policy. Then, policy improvement is performed to check if Tδ(s) that are computed are the same as corresponding Vδ(s). If they are the same for all states, the policy used to calculate Vδ(s) is optimal else the policy obtained from $T_\delta(s)$ is taken as the initial policy and the process is repeated until both $V_\delta(s)$ and $T_\delta(s)$ turn out to be the same.

For the present problem, a random policy of (Positive: Do Nothing; Neutral: Intervene; Negative: Intervene) was taken. After performing PI, ‘Do Nothing’ was arrived at as the optimal action for all the three states (for both commercial and consumer products). This methodology can be improvised by Lenovo by incorporating actual figures and a more proactive policy can thus be obtained.

Appendix D: Prediction equations

$pNPS_{Consumer} = 30.685 + 0.385*C1 + -1.379*C2 + 0.586*C3 + -0.528*C4 + -0.398*C5 + 0.397*C6 + -0.522*C7 + 0.917*C8 + -0.5*C9 + -1.42*C10 + -0.56*C11 + 0.877*C12 + -1.251*C13 + -1.9*C14 + 1.1*C15 + 2.598*C16 + -2.512*C17 + 0.894*C18 + 0.952*C19 + -1.818*C20 + -1.122*C21 + 1.793*C22 + 1.719*C23 + 1.369*C24 + 1.42*C25 + 1.209*C26 + 2.84*C27 + -1.095*C28 + -0.1*C29 + -1.33*C30 + -2.022*C31$

$pNPS_{Commercial} = 42.202 + -0.778C1 + 0.563C2 + -0.76C3 + 1.054C4 + 1.51C5 + 1.075C6 + -1.587C7 + 1.123C8 + -2.231C9 + -1.435C10 + 2.461C11 + -5.921C12 + -1.635C13 + 0.946C14 + 2.496C15 + 2.073C16 + -3.596C17 + 0.986C18 + -2.405C19 + 2.169C20 + 5.93C21 + 0.918C22$

Appendix E: Principal Component Regression data transformation

This section gives an example for transformation of data during PCR process. In reference to *Figure 1*, for Consumer Data, X is given by:

Table E1: Series-Month level X variables

			1	2	3		94	95	K = 96
			ACCESSORIES			…	CLIENT OS		
	Series	Month	NEGATIVE	NEUTRAL	POSITIVE	…	NEGATIVE	NEUTRAL	POSITIVE
1	A SERIES	May-17	5	15	2	…	5	10	3
2	A SERIES	Jun-17	0	7	4	…	1	5	0
3	A SERIES	Jul-17	8	6	1	…	4	0	2
4	A SERIES	Aug-17	1	9	2	…	0	2	3
5	A SERIES	Sep-17	5	3	3	…	4	2	1
	…	…	…	…	…	…	…	…	…
82	IDEAPAD 100 SERIES	Aug-17	18	21	8	…	8	27	11
83	IDEAPAD 100 SERIES	Sep-17	16	12	6	…	13	10	8
84	IDEAPAD 100 SERIES	Oct-17	14	16	13	…	10	24	11

The K variables are converted to A = 31 components after PCA and we obtain T:

Table E2: Series-Month level T variables

	Series	Month	C 1	C 2	C 3	…	C 29	C 30	C 31
1	A SERIES	May-17	-0.881	1.882	-0.819	…	1.077	-0.382	-0.119
2	A SERIES	Jun-17	-4.191	0.561	-0.318	…	-0.129	-0.379	0.981
3	A SERIES	Jul-17	-3.759	-0.107	0.046	…	0.384	-0.4	0.916
4	A SERIES	Aug-17	-3.266	-0.013	-0.884	…	-0.269	0.749	0.105
5	A SERIES	Sep-17	-3.082	0.385	0.294	…	-0.693	0.425	-1.213
	…	…	…	…	…	…	…	…	…
82	IDEAPAD 100 SERIES	Aug-17	2.922	6.78	-1.98	…	-0.438	1.809	-0.826
83	IDEAPAD 100 SERIES	Sep-17	2.093	5.478	-1.602	…	2.609	0.676	0.072
84	IDEAPAD 100 SERIES	Oct-17	4.219	7.737	-3.215	…	0.383	0.217	0.098

Table E3: pNPS values (y variable)

Month	pNPS
Oct-17	43.75
Nov-17	44.9
Dec-17	33.33
Jan-18	33.33
Feb-18	32.91
…	…
Jan-18	15.36
Feb-18	14.47
Mar-18	14.24

Appendix F: Example of Principal Components Regression for a Series-Month combination

Example: Yoga Series

In the Month o May 2017, X values are:

Table F1: X variables for Yoga Series

Tax #	Taxonomy	Sentiment	Frequency of sentiments
1	ACCESSORIES	NEGATIVE	84
2		NEUTRAL	92
3		POSITIVE	50
4	CLIENT OS	NEGATIVE	18
5		NEUTRAL	35
6		POSITIVE	14
…	…	…	…
94	GENERAL COMMENT	NEGATIVE	22
95		NEUTRAL	25
96		POSITIVE	122

After PCA, the T values we get for the month of May 2017 are:

Table F2: Component values

#	Component	Values
1	C1	23.163
2	C2	-0.789
3	C3	7.458
…	…	…
…	…	…
…	…	…
94	C 94	-0.309
95	C 95	0.038
96	C 96	-0.139

Substituting the T values of the variables onto the regression equation of *Consumer* Products:

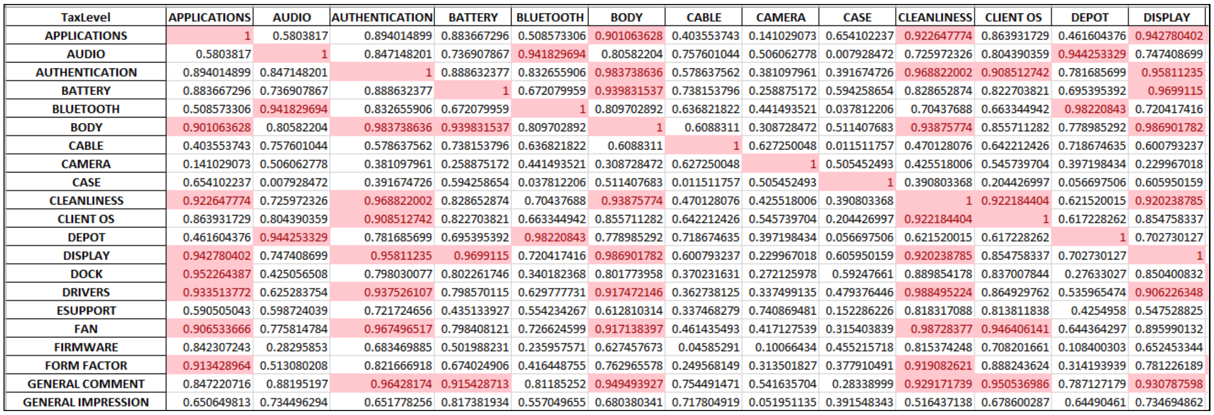
pNPS$_{\text{Consumer}}$ = 30.685 + (0.385)*C1 + (-1.379)*C2 +…..+ (-1.33)*C30 + (-2.022)*C31

We get, pNPS$_{\text{Consumer}}$ for YOGA series = 44.144.

This means, based on May 2017 data, we expect a pNPS for the month of October 2017 (5 month lag) to be **44.144**. From the survey data, we observe that the actual pNPS value for the month of October 2017 is **43.778**

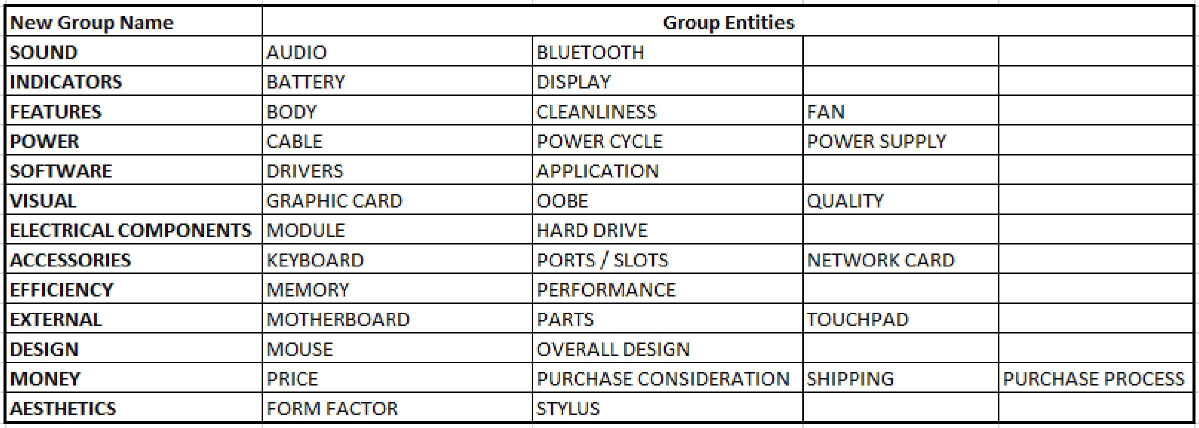
Appendix G: Selection of Variables

Grouping with the help of correlation matrix



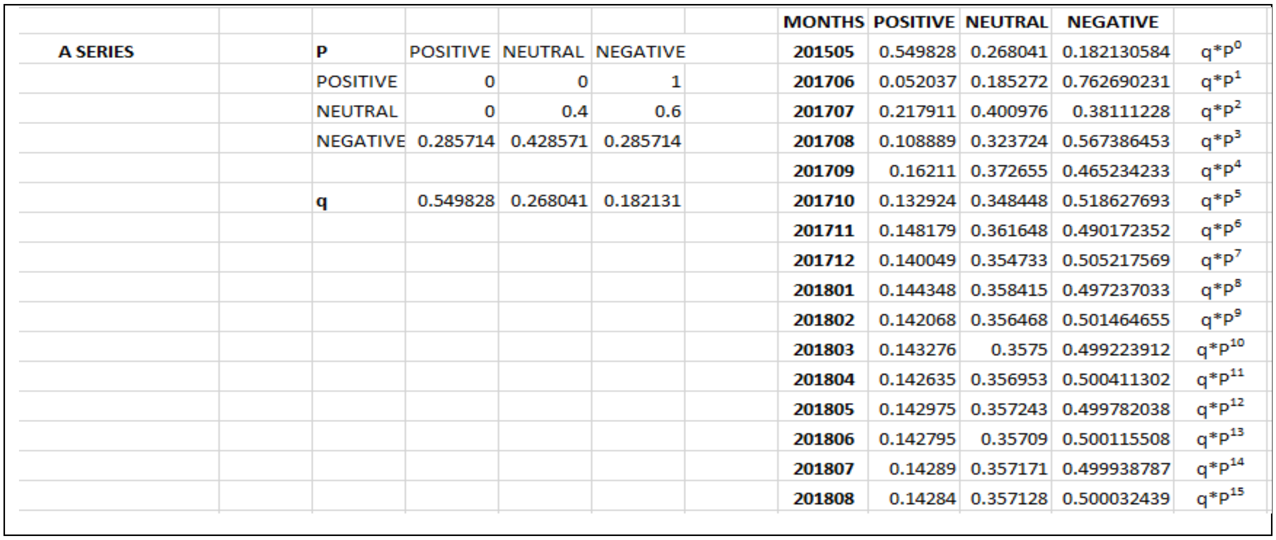
TaxLevel	APPLICATIONS	AUDIO	AUTHENTICATION	BATTERY	BLUETOOTH	BODY	CABLE	CAMERA	CASE	CLEANLINESS	CLIENT OS	DEPOT	DISPLAY
APPLICATIONS	1	0.5803817	0.894014899	0.883667296	0.508573306	0.901063628	0.403553743	0.141029073	0.654102237	0.922647774	0.863931729	0.461604376	0.942780402
AUDIO	0.5803817	1	0.847148201	0.736907867	0.941829694	0.80582204	0.757601044	0.506062778	0.007928472	0.725972326	0.804390359	0.944253329	0.747408699
AUTHENTICATION	0.894014899	0.847148201	1	0.888632377	0.832655906	0.983738636	0.578637562	0.381097961	0.391674726	0.968822002	0.908512742	0.781685699	0.95811235
BATTERY	0.883667296	0.736907867	0.888632377	1	0.672079959	0.939831537	0.738153796	0.258875172	0.594258654	0.828652874	0.822703821	0.695395392	0.9699115
BLUETOOTH	0.508573306	0.941829694	0.832655906	0.672079959	1	0.809702892	0.636821822	0.441493521	0.037812206	0.70437688	0.663344942	0.98220843	0.720417416
BODY	0.901063628	0.80582204	0.983738636	0.939831537	0.809702892	1	0.6088311	0.308728472	0.511407683	0.93875774	0.855711282	0.778985292	0.986901782
CABLE	0.403553743	0.757601044	0.578637562	0.738153796	0.636821822	0.6088311	1	0.627250048	0.011511757	0.470128076	0.642212426	0.718674635	0.600793237
CAMERA	0.141029073	0.506062778	0.381097961	0.258875172	0.441493521	0.308728472	0.627250048	1	0.505452493	0.425518006	0.545739704	0.397198434	0.229967018
CASE	0.654102237	0.007928472	0.391674726	0.594258654	0.037812206	0.511407683	0.011511757	0.505452493	1	0.390803368	0.204426997	0.056697506	0.605950159
CLEANLINESS	0.922647774	0.725972326	0.968822002	0.828652874	0.70437688	0.93875774	0.470128076	0.425518006	0.390803368	1	0.922184404	0.621520015	0.920238785
CLIENT OS	0.863931729	0.804390359	0.908512742	0.822703821	0.663344942	0.855711282	0.642212426	0.545739704	0.204426997	0.922184404	1	0.617228262	0.854758337
DEPOT	0.461604376	0.944253329	0.781685699	0.695395392	0.98220843	0.778985292	0.718674635	0.397198434	0.056697506	0.621520015	0.617228262	1	0.702730127
DISPLAY	0.942780402	0.747408699	0.95811235	0.9699115	0.720417416	0.986901782	0.600793237	0.229967018	0.605950159	0.920238785	0.854758337	0.702730127	1
DOCK	0.952264387	0.425056508	0.798030077	0.802261746	0.340182368	0.801773958	0.370231631	0.272125978	0.59247661	0.889854178	0.837007844	0.27633027	0.850400832
DRIVERS	0.933513772	0.625283754	0.937526107	0.798570115	0.629777731	0.917472146	0.362738125	0.337499135	0.479376446	0.988495224	0.864929762	0.535965474	0.906226348
ESUPPORT	0.590505043	0.598724039	0.721724656	0.435133927	0.554234267	0.612810314	0.337468279	0.740869481	0.152286226	0.818317088	0.813811838	0.4254958	0.547528825
FAN	0.906533666	0.775814784	0.967496517	0.798408121	0.726624599	0.917138397	0.461435493	0.417127539	0.315403839	0.98728377	0.946406141	0.644364297	0.895990132
FIRMWARE	0.842307243	0.28295853	0.683469885	0.501988231	0.235957571	0.627457673	0.04585291	0.10066434	0.455215718	0.815374248	0.708201661	0.108400303	0.652453344
FORM FACTOR	0.913428964	0.513080208	0.821666918	0.674024906	0.416448755	0.762965578	0.249568149	0.313501827	0.377910491	0.919082621	0.888243624	0.314193939	0.781226189
GENERAL COMMENT	0.847220716	0.88195197	0.96428174	0.915428713	0.81185252	0.949493927	0.754491471	0.541635704	0.28338999	0.929171739	0.950536986	0.787127179	0.930787598
GENERAL IMPRESSION	0.650649813	0.734496294	0.651778256	0.817381934	0.557049655	0.680380341	0.717804919	0.051951135	0.391548343	0.516437138	0.678600287	0.64490461	0.734694862

Figure G1: Correlation Matrix for consumer taxonomy (with 65 variables)



New Group Name	Group Entities			
SOUND	AUDIO	BLUETOOTH		
INDICATORS	BATTERY	DISPLAY		
FEATURES	BODY	CLEANLINESS	FAN	
POWER	CABLE	POWER CYCLE	POWER SUPPLY	
SOFTWARE	DRIVERS	APPLICATION		
VISUAL	GRAPHIC CARD	OOBE	QUALITY	
ELECTRICAL COMPONENTS	MODULE	HARD DRIVE		
ACCESSORIES	KEYBOARD	PORTS / SLOTS	NETWORK CARD	
EFFICIENCY	MEMORY	PERFORMANCE		
EXTERNAL	MOTHERBOARD	PARTS	TOUCHPAD	
DESIGN	MOUSE	OVERALL DESIGN		
MONEY	PRICE	PURCHASE CONSIDERATION	SHIPPING	PURCHASE PROCESS
AESTHETICS	FORM FACTOR	STYLUS		

Figure G2: Group names based on class and correlation for consumers



							MONTHS	POSITIVE	NEUTRAL	NEGATIVE	
A SERIES		P	POSITIVE	NEUTRAL	NEGATIVE		201505	0.549828	0.268041	0.182130584	$q*P^0$
		POSITIVE	0	0	1		201706	0.052037	0.185272	0.762690231	$q*P^1$
		NEUTRAL	0	0.4	0.6		201707	0.217911	0.400976	0.38111228	$q*P^2$
		NEGATIVE	0.285714	0.428571	0.285714		201708	0.108889	0.323724	0.567386453	$q*P^3$
							201709	0.16211	0.372655	0.465234233	$q*P^4$
		q	0.549828	0.268041	0.182131		201710	0.132924	0.348448	0.518627693	$q*P^5$
							201711	0.148179	0.361648	0.490172352	$q*P^6$
							201712	0.140049	0.354733	0.505217569	$q*P^7$
							201801	0.144348	0.358415	0.497237033	$q*P^8$
							201802	0.142068	0.356468	0.501464655	$q*P^9$
							201803	0.143276	0.3575	0.499223912	$q*P^{10}$
							201804	0.142635	0.356953	0.500411302	$q*P^{11}$
							201805	0.142975	0.357243	0.499782038	$q*P^{12}$
							201806	0.142795	0.35709	0.500115508	$q*P^{13}$
							201807	0.14289	0.357171	0.499938787	$q*P^{14}$
							201808	0.14284	0.357128	0.500032439	$q*P^{15}$

Figure G3: Calculation of EOS

Appendix H: Areas of strength and improvements

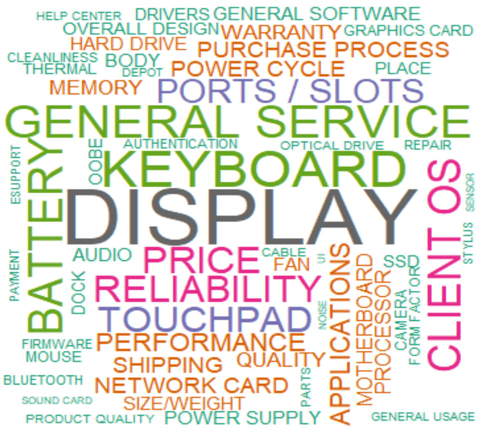


Figure H1: Negative sentiment

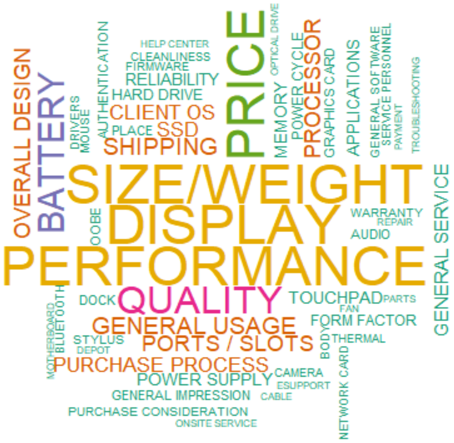


Figure H2: Positive Sentiment

References

1. Penn State College of Science - Statistics: https://onlinecourses.science.psu.edu
2. Process improvement using data - https://learnche.org/pid/
3. NCSS Statistical Software documentation - https://ncss-wpengine.netdna-ssl.com/
4. Wikipedia
5. Applied Statistics & Probability for engineers (Authors- Douglas C. Montgomery, George C. Runger)
6. Value iteration and policy iteration algorithms for Markov decision problem - Elena Pashenkova, Irina Rish, Rina Dechter (Department of Information and Computer Science, University of California at Irvine CA 92717)