2012

Determining optimum insurance product portfolio through predictive analytics

BADM Final Project Report

Dinesh Ganti(61310071), Gauri Singh(61310560), Ravi Shankar(61310210), Shouri Kamtala(61310215), Supreet Kaur(61310595) Vinayak Palankar(61310074)
Section A Group 1
12/27/2012

Table of Contents

EXECUTIVE SUMMARY

Business Problem – An entrepreneur has an idea for a new business venture, which, in a nutshell, is to offer insurance to customers on price drops of certain products. Registered customers get a certain multiple of the insurance fee or a certain percentage of the drop provided that the drop happens within a certain period after purchase. Insurance is offered on a new portfolio of products every day and, for the sake of simplicity, the period within which price needs to drop is taken as one day. The business problem is to understand from the data, if the business idea is theoretically and financially viable.

Data – The available data is a list of products sold across five e-commerce sites. The data gives us information about shipping, average ratings, brand, category, in stock etc. Some features of the data are (a) Categorical variables such as model name, brand etc. which cannot simply be converted as they would explode the number of categorical variables (b) columns like average rating and review count which had large number of missing values and were filled through imputation based on rest of the columns (c) textual data columns such as shipping period which were converted into numerical averages. Data such as this is available on the internet and can be crawled off various sites easily.

Data Analytics solution – To check if it is possible to construct a portfolio of products such that as high a percentage of the products selected in the portfolio are due for a price increase so that losses on price drops don't overwhelm the insurance fee gained where the price goes up or stays the same. The idea is to build various classification models that can predict if the price will go up or not and then construct a portfolio by selecting top **x**% of products that have the highest predicted probability that the price will go up next day among all the products considered. Observe that data mining only takes us towards making a decision and quite a bit of external information, analysis and assumptions are needed before one can come to a conclusion. To achieve the data mining goal, we would (a) Use available data of various products sold by the five online retailers as predictors to classify products as 'Price Up' or not (b) Rank order the results in terms of probability of price increase (c) Select top x% to offer insurance on.

Recommendations – Our data analysis shows that if products are randomly selected to form a portfolio only 2.4% of products on an average would show a price increase. However, using our best model of Classification tree, this percentage can shoot up to as high as **100%** based on what "**x**" is chosen. If the top x% is intelligently and variably chosen everyday **and** if the multiple on insurance fee paid to customers upon price drop is carefully chosen so that it doesn't overwhelm the insurance revenue, the business can be very much viable. The entrepreneur can always explore additional streams of revenue like advertisements on the site or building a user database.

PROBLEM DESCRIPTION (BUSINESS AND ANALYTICS)

A young and bright entrepreneur has a new business idea. Customers are usually left disgruntled if there is a price drop on products they have recently purchased. More recent the purchase, more disgruntled one can expect customers to be. The idea is to offer insurance to customers on price drops. So, in case, the price drops within a pre-specified time, the customer is paid a multiple of insurance fee collected or a certain percentage of the price drop. Customer forfeits the insurance fee if the price goes up or remains the same within the period. The scope is limited to electronic products purchased on e-commerce websites. We are playing the role of consultants to this entrepreneur.

Every insurance company works on the principle of minority defaults. Similarly, in this case, insurance must be offered on a set of products such that the actual price drop compensations do not exceed the total insurance token amounts collected.

The business problem is to understand if the model is viable and advise the entrepreneur accordingly. As consultants, we would also help the entrepreneur get an idea regarding the number of products he/she needs to pick up to form a portfolio on a day to day basis and also what sort of a multiple he can offer on the insurance fee originally paid in case of a price drop.

The analytics goal is to mine the available data and see if one can come up with a classification model that can accurately predict price increases on products to such an extent that the product portfolio formed on the models predictions turns an overall profit for the entrepreneur.

DATA DESCRIPTION

Data has been collected from five e-commerce sites namely BuyThePrice, HomeShop18, Indiaplaza, Infibeam and Saholic. Information such as model, brand, color, price, shipping period, stock-out and whether or not the price has increased is available on various products offered by these sites.

Data preparation for building the model included:

- Converting categorical data into dummies. E.g. freeShipping, siteName and InStock. For running the Naïve Bayes method, we would need them as categories but for K-Nearest Neighbours, we needed to create categorical dummies. Dummies were also created for Brand and Color variables - since there were many different brands and colors, they were grouped first based on the outcome variable (brands/colors that have a very close average in terms of the output variable were put together as one group) and then categorized through dummy variables. These variables ultimately don't feature in any of the models as they didn't seem too important from a

price up or not perspective and they were making the models too complicated with little improvement in predictive accuracy.

- Predicting missing values of input variables using K-NN. E.g. avRating and reviewCount. These variables had many values missing. Some of the other predictors available such as brand, color, free shipping, web portal etc. were used to predict the average rating or the review count where this information was missing.
- Numerical transformation of textual data using averages. E.g. shippingPeriod. All the text had to be filtered out and the mid-value within the range specified in the test was used here. There were a few missing values in this variable as well but they were filled in using the average value of the shipping period across all products.
- Derivation of values from existing variables. E.g. timeLast from date
- Some columns such as name and group were ignored

Please refer to the **Exhibit 0** for a few data visualizations.

DATA MINING SOLUTION

K-Nearest Neighbors The original dataset given has 7,765 rows which were partitioned into training and validation datasets. We also had a separate hold-out set on which we tested the model once we built it. The best "k" that minimizes error in the validation set turned out to be K = 5. Please refer table below.

Value of k	% Error Training	% Error Validation	
4	1.59	2.25	
5	1.63	2.03	<--- Best k
6	1.63	2.06	

Please refer to **Exhibit 1A** for the classification matrices on the training and validation datasets. Please refer to **Exhibit 1B** for the misclassification matrices based on the naïve rule. The KNN model built was used to score the test data. The lift chart is the ideal metric to use since we are concerned with picking up the top x% of products. The lift chart obtained for KNN on validation and test data is in **Exhibit 1C**.

The following table demonstrates how one must select "x" based on performance on test dataset.

# Products	Cumulative PriceUp using average	Cumulative PriceUp when sorted using predicted values	% of PriceUp cases
12	0.583	8	66.67%
13	0.631	9	69.23%
14	0.680	10	71.43%
15	0.729	10	66.67%
16	0.777	10	62.50%

Once products are sorted in descending order of predicted probabilities from the model based on test data, the number of products must be chosen such that the percentage of products where the price has actually gone up is highest. For example, in the above case, the number of products that should be chosen to offer insurance on is 14 out of a possible 350 considered in the test dataset or top 4% of products. We recommend that all the price change data from the previous day be used to retrain and validate the models every day. The revised models should then be run on the next day's consideration set of products and a similar top x% based on predicted probabilities must be picked up to offer insurance on. The single most important decision criterion among the various models is the identification of a top x% of products that has the **maximum** number of possible price increases that happened the last day. Data of price increases must be crawled and the modeling exercise done every day to keep the models fresh. Once "x" is determined, the percentage of products that might show a price rise is known and the return on insurance fee paid in case of price drop can be determined. For example, in the above case, since ~70% of the products chosen are likely to show a price increase, the entrepreneur may, at worst, end up paying a multiple of insurance fee on only 30% of products chosen. We say "at worst" because some products among the 30% will not show a price change and no money is lost on them. The insurance multiple that can be safely offered then in 70/30 = 2.33 (assuming simply that an equal number of people purchased insurance on all the products in the portfolio). Please note that the insurance multiple must be changed every day.

Naïve Bayes The lift charts based on running this method on the test data set can be seen in **Exhibit 2A**. The table below shows why this model was not chosen as best. The best possible percentage is 41.18% if top 17 of 350 or top 4.86% of the products are chosen (not comparable to 71.43% from KNN)

# Products	Cumulative PriceUp using average	Cumulative PriceUp when sorted using predicted values	% of PriceUp cases
15	0.729	5	33.33%
16	0.777	6	37.50%
17	0.826	7	41.18%
18	0.874	7	38.89%
19	0.923	7	36.84%

Classification Tree The lift charts based on running this method on the test data set can be seen in **Exhibit 3A**. The table below shows why this model is the best. The best possible percentage is 100% if top 10 of 350 or top 2.86% of the products are chosen (compared to 71.43%-KNN and 41.18%-NB.)

# Products	Cumulative PriceUp using average	Cumulative PriceUp when sorted using predicted values	% of PriceUp cases
8	0.388571429	8	100.00%

9	0.437142857	9	100.00%
10	0.485714286	10	100.00%
11	0.534285714	10	90.91%
12	0.582857143	10	83.33%

Ensembles The only motivation to try ensembles after achieving 100% would be to increase the number of products on which to offer insurance and thus widen the net. We looked at a weighted combination of the probabilities obtained from the 3 models described above but we were not able to increase the number of products to offer insurance on. Weights were decided as inverse ratio of the error percentages on the test dataset. We end up with the exact same table above even for ensembles.

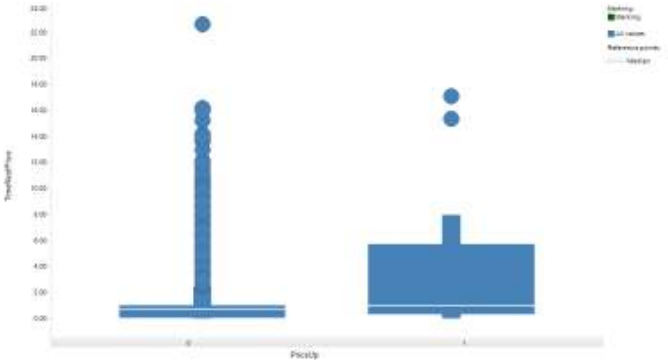
CONCLUSIONS AND OPERATIONAL RECOMMENDATIONS

- **Challenges in data collection**: The model predicts price up and price down based on product model, brand, shipping and other meta-information. Collecting such data would require a web-crawler engine which parses these web sites dynamically and captures all the required meta-tags. The crawling software has to be built or acquired. Data needs to be collected **daily**.
- **Data-driven**: It might be tempting as an insurer to deviate from the product portfolio suggested by the model and include products where there is a general expectation of price increase. While such manual massaging works in the short term, it is a non-scalable approach. Automatic, data-driven portfolios are the way to go and for this. Time and money must be invested on improving data sources and data collection.
- **Legal bindings**: One should ensure that the insurance licenses cover the offered products. Also, the web crawling approach could be sued by the e-commerce sites and this must be taken into account in the planning stages.
- **Review**: The models and the web-crawling software should be reviewed and revised **daily**.
- **Deployment:** At the end of a day, all instances of price changes/non-changes must be captured and the models retrained, validated and tested on this data. Once finalized, the models need to be run on the consideration set for the next day and the x% and multiple details need to be figured out based on the test data set performance of the models. It is quite possible that the same model doesn't come out on top every day. If insurance is being offered on price drops over a certain period rather than a day, the problem becomes a lot more complex and historical information will need to be tracked.

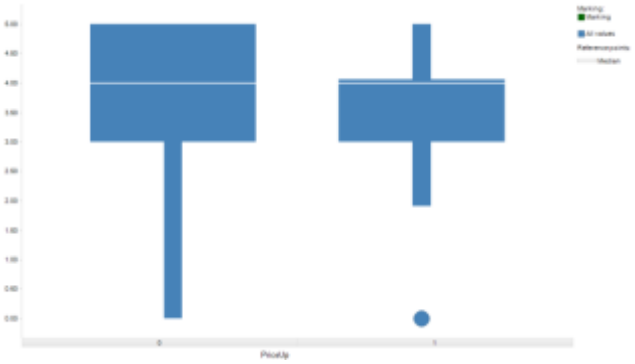
APPENDIX

Exhibit 0: Data Visualizations

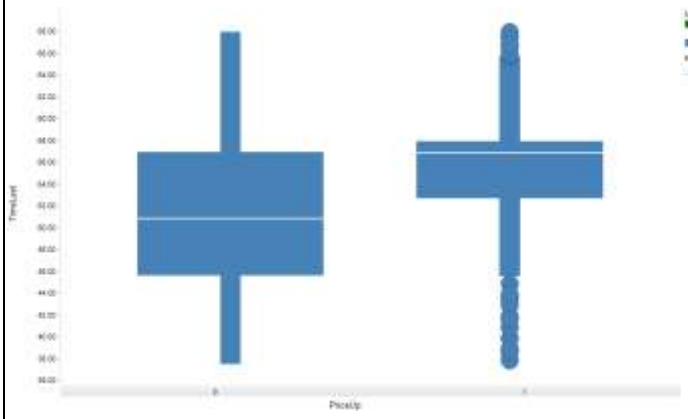
Effect of list price on the PriceUp variable



Effect of average rating on the PriceUp variable



Effect of TimeLast on the PriceUp variable



Effect of TimeNextPrice on the PriceUp variable

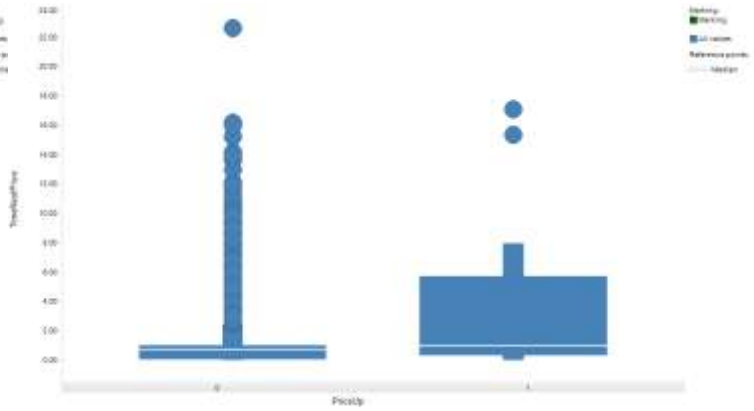


Exhibit 1A: Performance of KNN on the training and validation data

Training Data scoring - Summary Report (for k=5)

Cut off Prob.Val. for Success (Updatable)	0.5

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	37	71
0	5	4546

Error Report			
Class	# Cases	# Errors	% Error
1	108	71	65.74
0	4551	5	0.11
Overall	4659	76	1.63

Validation Data scoring - Summary Report (for k=5)

Cut off Prob.Val. for Success (Updatable)	0.5

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	21	58
0	5	3022

Error Report			
Class	# Cases	# Errors	% Error
1	79	58	73.42
0	3027	5	0.17
Overall	3106	63	2.03

Exhibit 1B: Performance of Naïve rule on the training and validation datasets created for KNN

Training Data scoring - Naïve Rule

Classification Confusion Matrix		
	Predicted Class	
Actual	1	0
1	0	108
0	0	4551

Error Report			
Class	# Cases	# Errors	% Error
1	108	108	100.00
0	4551	0	0.00
Overall	4659	108	2.32

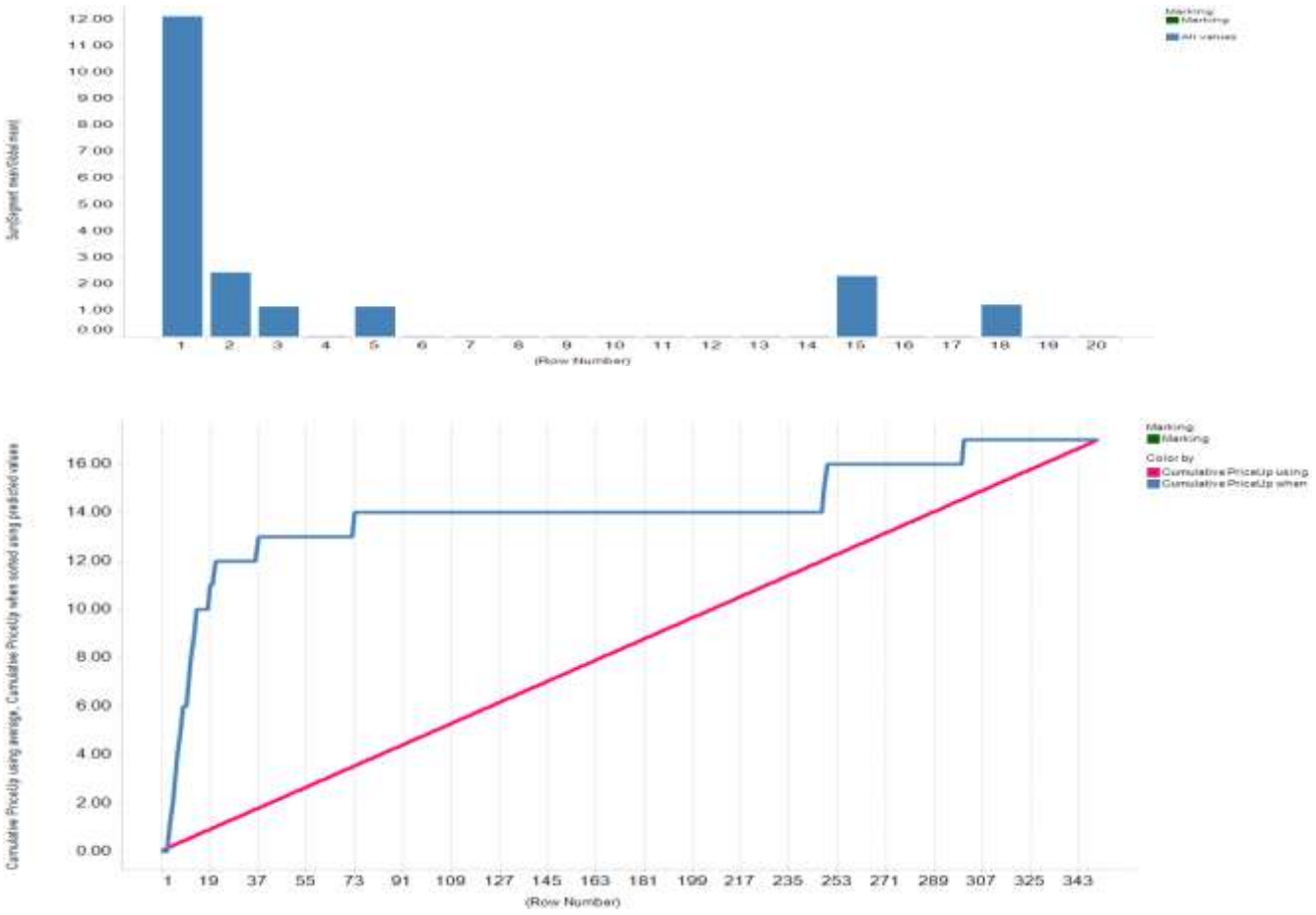
Validation Data scoring - Naïve Rule

Classification Confusion Matrix		
	Predicted Class	
Actual	1	0
1	0	79
0	0	3027

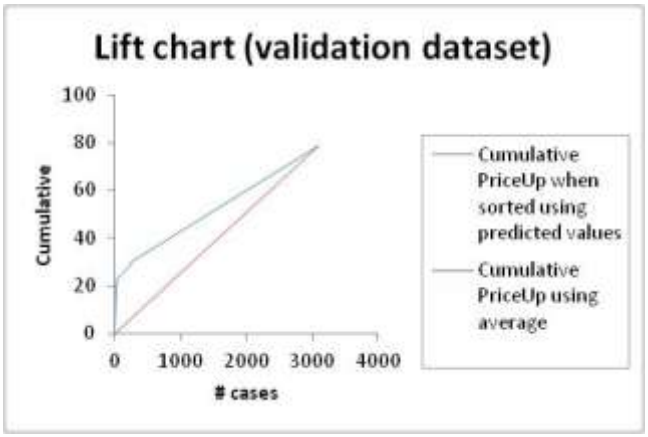
Error Report			
Class	# Cases	# Errors	% Error
1	79	79	100.00
0	3027	0	0.00
Overall	3106	79	2.54

Exhibit 1C:

Lift charts for KNN on test data



Lift Charts of KNN on validation data



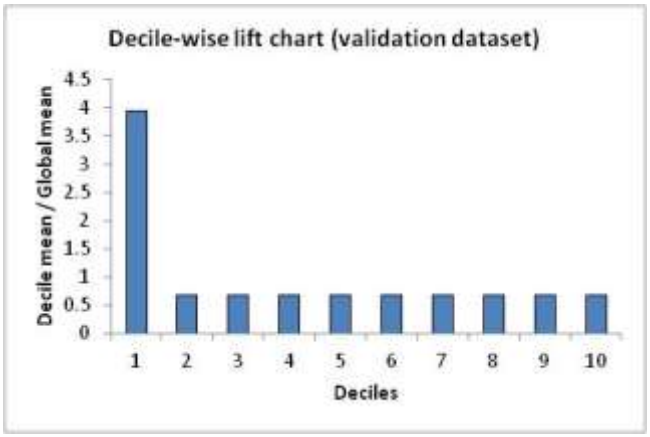
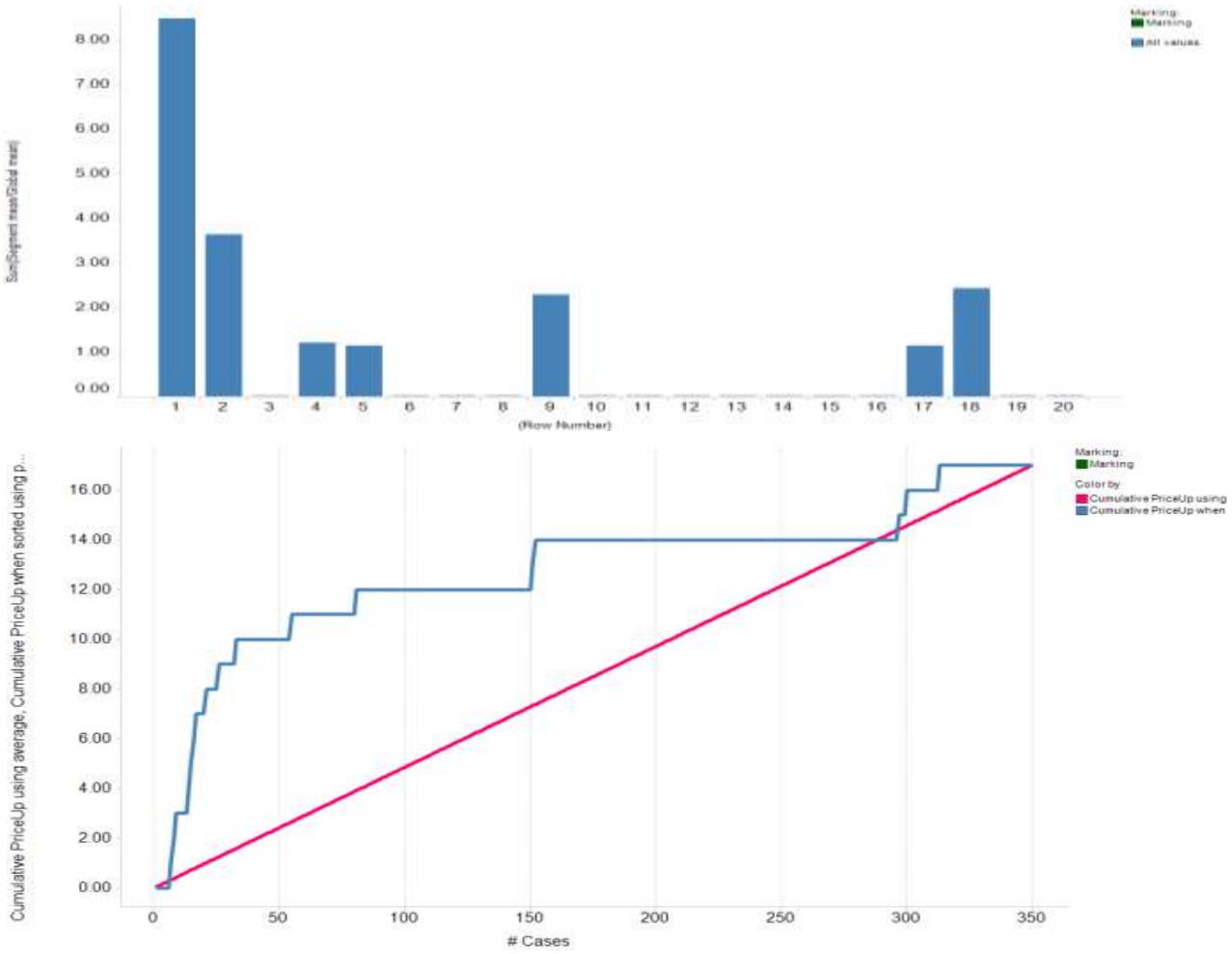
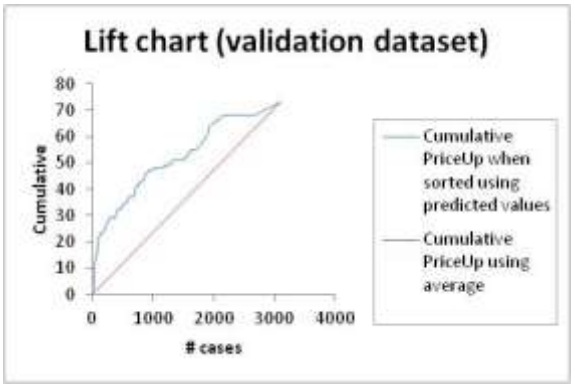


Exhibit 2A: Lift charts on the test dataset based on Naïve Bayes



Lift Charts of Naïve Bayes on validation data



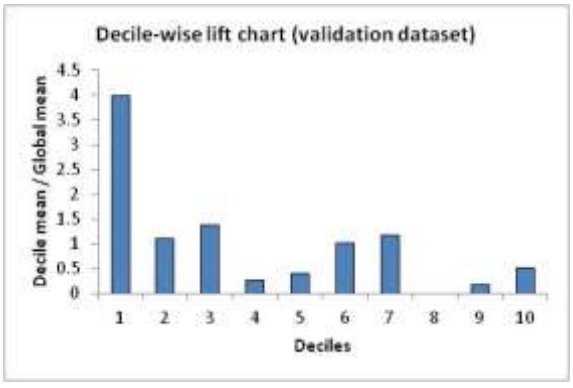
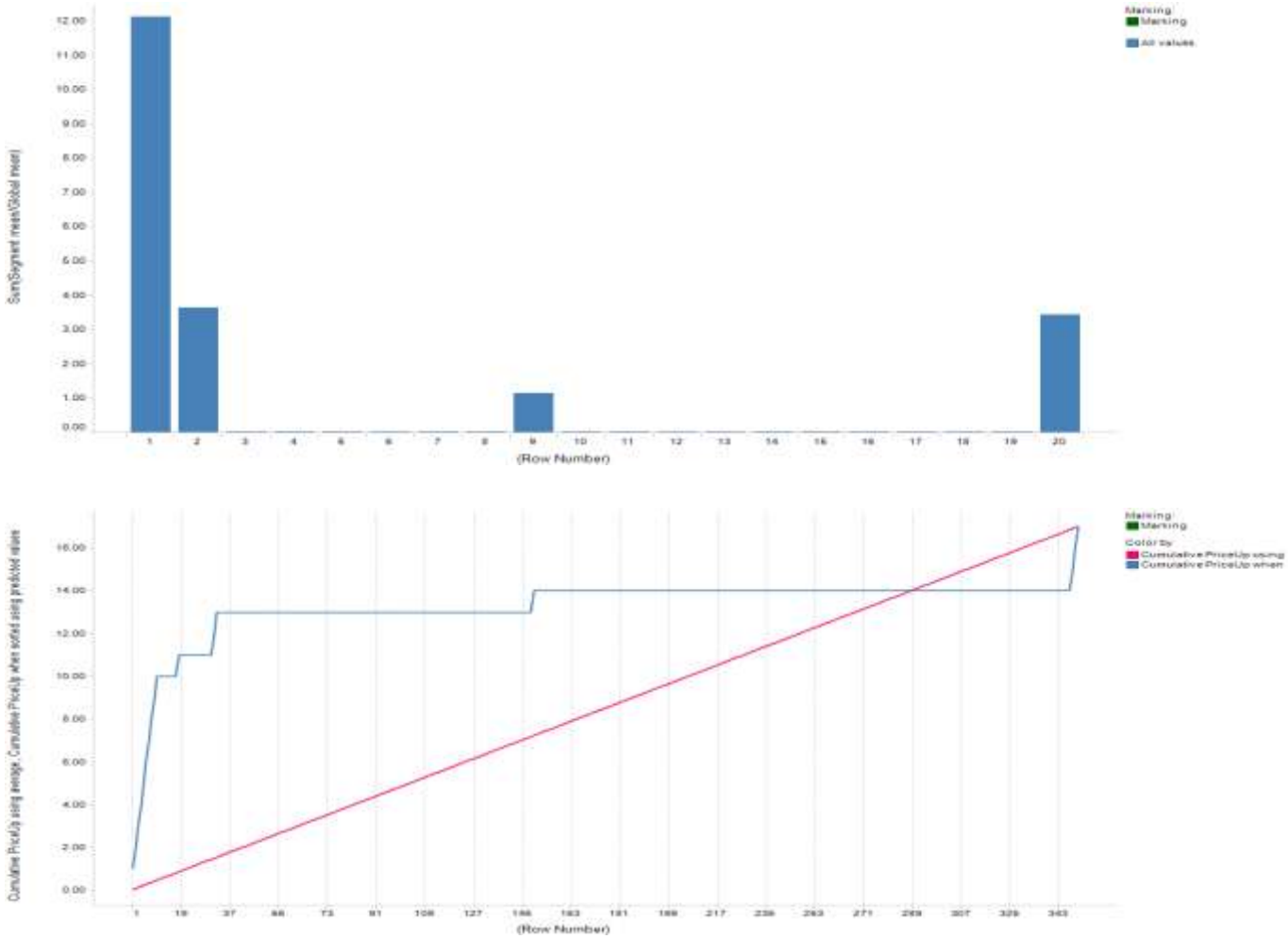


Exhibit 3A: Lift charts on the test dataset based on Classification Tree



Lift Charts of Classification Tree on Validation dataset