MACHINE LEARNING FOR RETAIL PRICE OPTIMIZATION

Student's name: Luong Do Van Quyen
Student's ID: MAMAIU18057
Thesis Supervisor: Tran Vu Khanh, Ph.D

Contents

1. INTRODUCTION

- Vietnam is one of the world's most appealing Food And Beverage (FB) markets, according market research company BMI [0].
- Traditional pricing is generally not sensitive enough to accurately evaluate the influence of various factors.
- Hence, finding an optimal price point by machine learning for all the products is a critical business need which maximizes the overall revenue.
- We collect data from a Burger Cafe of Microsoft building in China, based on Xueshan Zhang's Research [**?**].
- We divide process into three parts: The Prediction Model, Elasticity Of Price, The Price Optimization.

1. INTRODUCTION

The Prediction Model uses the following theoretical models:

- Linear Models:
  1. Linear Regression
- Tree-based Methods
  1. Decision Tree
  2. Random Forest
  3. XGBoost

The evaluation of the above models are measured with the below metrics:

- Mean Square Error
- Mean Absolute Error
- Root-mean-square error
- R^2

Then, using Elasticity Of Price and linear programming to find The Price Optimization.

1. INTRODUCTION

Literature reviews

- K-means clustering and regression model were used by Rajan Gupta and Chaitanya Pathak in 2014. With the assistance of past user cluster and price range data, this model provided a price range at the level of the user cluster.
- In 2013, Kyle Christianson and James Fuller, applied price optimization method to determine price for room in hotel, based on market (competitor) prices, room availability, predicted demand (based on historical data on hotel visitors' length of stay, rate segment, day of the week, and lead time), and business rules.
- For the Airbnb platform's hosts, authors used a regression model to forecast the best pricing for each offering for a given night.

The Linear Models could be described by the regression line:

$$Y_t = \alpha + \beta X_t + \mu_t, \forall\, t = \overline{1, n}$$

where α is the intercept, β is coefficient or slope, μ is random disturbance term, X is the independent variable and Y is the dependent variable.

The least squares method chooses $\hat{\alpha}$ and $\hat{\beta}$ that minimize the loss function defined as

$$L = \sum_{i=1}^{n} \hat{u_i}^2 = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n} (Y_i - \hat{\alpha} - \hat{\beta} X_i)^2.$$

Building a regression tree:

1. The predictor space is divided into J distinct regions, $R_1, R_2, ..., R_J$ that minimize the loss function defined as

$$L = \sum_{i=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{R_j}$ is the mean value of the training observations within that j^{th} region.

2. The prediction f of every testing observation that falls into the region R_j is $\hat{y}_{R_j}$.

Random Forest Algorithms:

1. Sample with replacement b training samples from N: $N_b(X_b, Y_b)$.
2. Train a regression tree f_b on X_b, Y_b. When building decision trees, a random sample of m attributes is chosen from the full set of p attributes.
3. After training, predictions for test samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x').$$

XGBoost formula:

$$\gamma_m = argmin \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)),$$

$$F_m(x) = F_{m-1}(x) + \nu_m h_m(x).$$

L : loss function

γ : predicted value.

y_i : observed value

Metrics to evaluate models

$$MSE = \frac{1}{T - (T_1 - 1)} \sum_{t=T_1}^{T} (y_{t+s} - f_{t,s})^2,$$

$$MAE = \frac{1}{T - (T_1 - 1)} \sum_{t=T_1}^{T} |y_{t+s} - f_{t,s}|,$$

where T_1 is the first out-of-sample forecast observation and T is the entire sample size (in-sample + out-of-sample).
And RMSD (Root-mean-square deviation) or RMSE (Root-mean-square error) is calculated by: $\sqrt{MSE}$.

Price elasticity of demand :

$$Ed = \frac{\Delta Q}{Q} . \frac{P}{\Delta P},$$

where Q, P are the original demand and price for the product, respectively. ΔQ represents the change in demand as the price changes, and ΔP depicts the change in the product's price.

$$max \sum_{k=1}^{3} \sum_{i=1}^{n} p_{ik} x_{ik} d_{ik},$$

$$w.r.t$$

$$\sum_{k=1}^{3} x_{ik} = 1 \quad \forall\, i = \overline{1, n},$$

$$\sum_{k=1}^{3} \sum_{i=1}^{n} p_{ik} x_{ik} = c,$$

$$x_{ik} \epsilon \left\{0, 1\right\}.$$

In which, p_{ik} represents the price and d_{ik} represents the demand for the i^{th} product if the k^{th} price-demand combination is selected.

We collect data from a Burger Cafe of Microsoft building in China, based on Xueshan Zhang's Research [**?**].

1. Cafe-Transaction-Store: The dataset contains statistics on the selling price and quantity of burger items in the store by day.
2. Cafe-Sell-Meta-Data: This dataset describes information about burgers and burger combos in the store.
3. Date-Infodata: This set contains calendar information and other data such as holidays, weekends,...

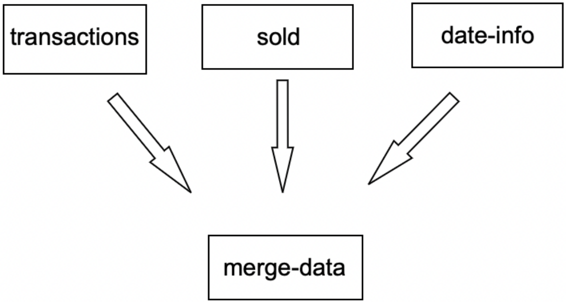

Figure 1: Combining data sets

We need univariate and multivariate analysis to select the data sets that fit the model and see how they affect our model. This is a very important step in building a machine learning model.

- Univariate: days off, weekends, holidays, and no-go days less than normal days.
- Multivariable: on holidays, schoolbreaks, weekends the purchase quantity will be less, the temperature has little effect on the quantity demanded.

We will divide dataset into 3 main subdata sets. In which:

- Data merge is the original data (consider variables PRICE and QUANTITY);
- Data bau will consider PRICE and QUANTITY in special factors: No Holiday, No weekend, No schoolbreak, People go out;
- Data full is a model that will use factors other than PRICE (including outdoor, weekend, schoolbreak) to predict QUANTITY, with the desire to find the optimal price in all conditions, without need to be considered separately for special cases.

Type	Linear Regression	Random Forest	XGBoost	Decision Tree
MAE	0.586	0.377	0.373	0.373
MSE	0.509	0.226	0.227	0.227
RMSE	0.713	0.476	0.476	0.476
R^2	0.118	0.606	0.606	0.606

Table 1: Evaluate each model in Data merge

On the first 80% of historical data, all models were trained. The previous 20% data were utilized for the test data.
Random Forest, XGBoost and Decision Tree gave better resutls than Linear Regression. Here, the results of the R-squared are quite low, the accuracy is just under 60% in all models.

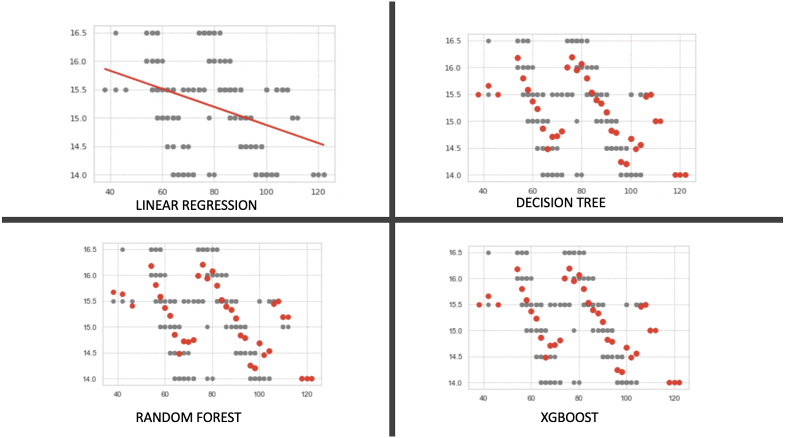

Figure 2: Data merge

The data is quite messy in the models.

Type	Linear Regression	Random Forest	XGBoost	Decision Tree
MAE	2.525	2.426	2.425	2.425
MSE	8.520	8.253	8.242	8.242
RMSE	2.918	2.872	2.870	2.870
R^2	0.817	0.823	0.823	0.823

Table 2: Evaluate each model

We have better results in table 2 for R^2 than results of Data merge in table 1. Most of which are above 80%.

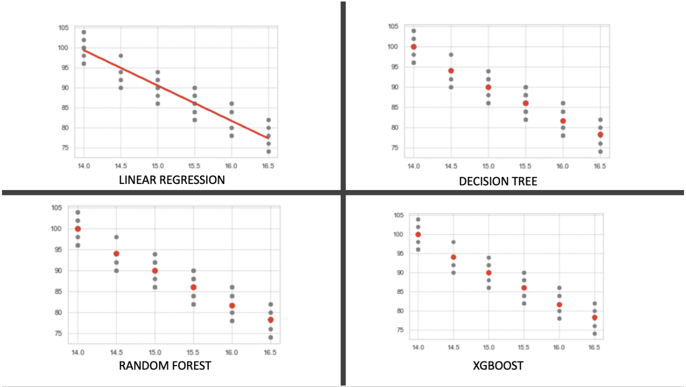

Figure 3: Data bau

In particular, the data is almost linear and fits the linear regression model. The indicators that evaluate this model are also very good.

Type	Random Forest	XGBoost	Decision Tree
MAE	4.600	4.589	4.589
MSE	57.896	57.845	57.845
RMSE	7.608	7.605	7.605
R^2	0.789	0.789	0.789

Table 3: Evaluate each model

We get R^2 results in table 3 which are also quite good in the models. However, R^2 in this table are less than Table 2 (Data bau).

In our goal of building a product's price elasticity, we will use linear regression model with outliers removed (linear regression of data bau). Because according to the results considered in tables 1, 2, 3, this model has the best results of R^2. So we'll use it to find price elasticity. We will consider each product separately by its SELL_ID (1070, 2051, 2052, 2053) and get metric scores in table 4.

Type	MAE	MSE	RMSE	R^2
SELL_ID = 1070	2.525	8.520	2.918	0.817
SELL_ID = 2051	2.563	8.746	2.957	0.369
SELL_ID = 2052	2.312	7.455	2.703	0.414
SELL_ID = 2053	2.443	8.120	2.849	0.811

Table 4: Evaluate each SELL_ID

With price elasticity lines $q_{QUANTITY} = a.p_{PRICE} + b$, we calculate the price elasticity (EpD) at each value point on the line, where a, b are in OLS Regression Result. Then, we have following results:

- Linear model of SELL_ID 1070: $q = -8.83p + 222.93$
- Linear model of SELL_ID 2051: $q = -3.60p + 75.93$
- Linear model of SELL_ID 2052: $q = -2.55p + 52.43$
- Linear model of SELL_ID 2053: $q = -5.89p + 120.67$

SELL_ID 1070 will have the biggest change in quantity sold if the store changes its selling price.

Apply the formula $profit = (sellingprice - cost) \times quantity$ with quantity from Elasticity Of Price. We fix the cost of making a product at \$9 because we find the minimum price in this data is \$10.12 and maximum is \$16.5. Then, we set the selling price from the lowest current selling price of -1 (here we use \$9.12) to the lowest current selling price of +10 (\$20.12). We get the following output for each SELL_ID:

Type	**Price**	**Quantity**	**Profit**
SELL_ID = 1070	17.1	72.009	583.273
SELL_ID = 2051	15.07	21.709	131.776
SELL_ID = 2052	14.82	14.711	85.618
SELL_ID = 2053	14.75	33.810	194.410

Table 5: Optimal price for each SELL_ID

5. CONCLUSION AND DISCUSSION

- We studied the algorithms of Linear and Tree-based models.
- We tried try to clearly show how to conduct data analysis and build models as a basis and premise.
- We realized Linear regression measured by R^2 is still a good and data-friendly method to apply to price elasticity content.
- We understood how to demand forecasting is essential to develop the following steps.

5. REFERENCES

[1]Vivek Kumar Singh and Kaushik Dutta. 2015. Dynamic Price Prediction for Ama- zon Spot Instances. 2015 48th Hawaii International Conference on System Sciences (2015), 1513–1520.

[2]Rainer Schlosser and Martin Boissier. 2018. Dynamic pricing under competition on online marketplaces: A data-driven approach. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining. ACM, 705–714.

[3]Rajan Gupta and Chaitanya Pathak. 2014. A Machine Learning Framework for Predicting Purchase by Online Customers based on Dynamic Pricing. In Complex Adaptive Systems.