CS148 Final project Report

Contributors:

Naibiao Jin Student ID: 305856171

Zhaoheng Luo Student ID: 705859484

Contents

1 Executive Summary

Part A: Predictive model for sales by brand

We built two features on sales (sales in previous month and rolling average of 3-Month sales), one feature about the brand (Category L1), and extract the time information by creating three time-related features: year, month, quarter. We then construct three models (Linear Regression, Random Forest and XGBoost model) to run test on. The result is that our best model, XGBoost does well in predicting the sales of big brand (monthly total sales>500,000) and medium brand (monthly total sales<500,000 and >10,000) with a R^2 score of 0.993 and 0.786 respectively.

Part B: Finding key indicators for successful product

We build two metrics to measure how good a certain category of product sales in the market: market share score and out performance score. Market share score shows the empirical probability a certain product that a certain product's sales will rank in the top third of the market. Out performance score shows how much a certain indicator can affect the market share score. After traversing all possible values for all the features given in the dataset, we find that the "Flower" in Category L2 and "Hybrid" in Category L3 is the best sales-prompt indicators.

This result tells us that picking a growing, less-competitive niche market, is the key to grow revenue in the cannabis industry.

2 Background/Introduction

2.1 Regulation policy

Obviously, the main reason for the restricted sales of the cannabis industry is not production capacity, but legal regulatory policies. For example, in some states in the United States (e.g. CA, CO, MI) cannabis is completely legalized and can be used by adults. Some states (e.g. PA, OK, FL) have not fully liberalized the restrictions on cannabis use and only allow medical use. We can see that the current regulation on cannabis sales is loosening and the market continues to grow. Therefore, one of the most important factors that should be considered in predicting the sales of the cannabis market is the government's policy on cannabis, which is the anticipated emerging market. If at a certain period of time, the policy determines that adults in certain areas can legally use cannabis, then the demand in these areas will surely inject great vitality into the entire cannabis market and predictably increase cannabis sales. Therefore, in terms of sales forecasting, judging market trends by paying attention to regulatory policies is a useful consideration.

2.2 Diversity of sales channels and accessibility of products

In the case of market demand, how to respond to the demand and truly convert market demand into actual sales? One of the important determinants is the capacity of sales channels, such as the number of dispensers, online shops, government-operated stores and private shops. The new market demand needs sufficient sales channels to meet. The distribution of sales outlets will also affect consumers' experience in buying cannabis. Some consumers may feel that it is inconvenient to buy cannabis and therefore give up buying them, which would affect the sales of cannabis to a certain extent.

2.3 Newly launched products

In the process of forecasting sales, because some products are newly launched, their historical sales data are few, and there is no reliable basis to predict their future sales. There are many variables, such as changes in consumer attitudes towards new products, and product quality issues. How to use limited data to deal with the prediction of new products is a big challenge. For example, we cannot give too low weight to new products, because this may underestimate the market potential of new products, and its sales may increase wildly, which greatly affects the total market sales.

2.4 Wide variety of products

There are many subdivisions of cannabis products, with different formats or types. If you want to accurately predict the sales of cannabis market, it is best to make predictions by category. However, due to the existence of too many types, the prediction problem will become very complicated, considering the feature dimensions of the predictive model. And how to collect data on various products is also a problem.

2.5 Changing unit pricing

Companies in the cannabis industry usually dynamically adjust the unit price of products based on market conditions. For example, retailers may develop category and brand cross-selling strategies or some promotional plans. In these scenarios, the selling price of each cannabis product is also volatile, and difficult to predict. Most customers are price-sensitive, and price changes are most likely to affect customer purchases and thus the sales of the entire cannabis market. Therefore, we may need to make more efforts and come up with more methods to overcome this problem.

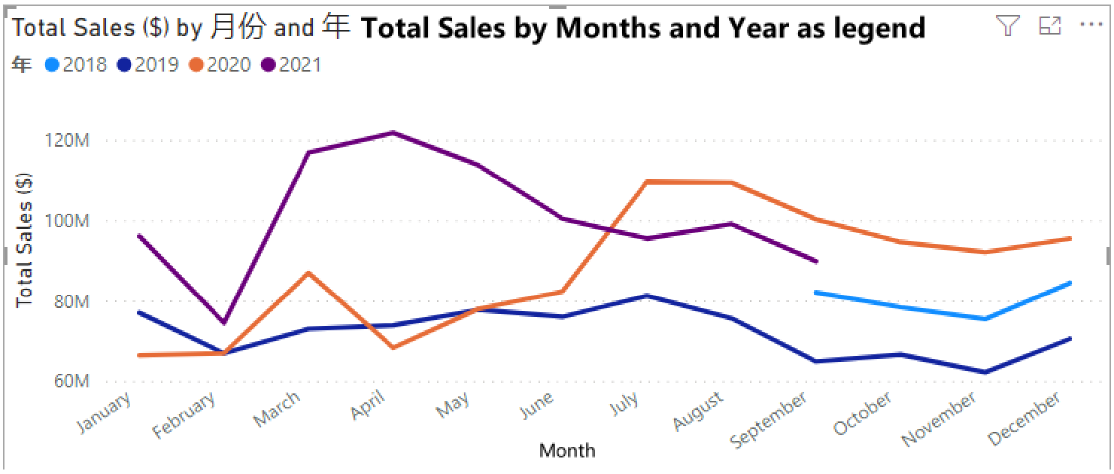
3 Methodology

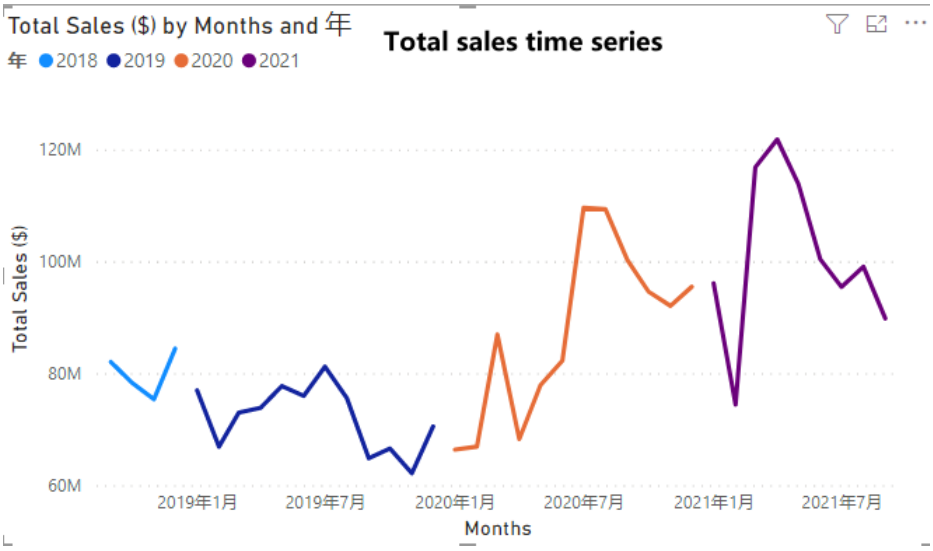
Part A: Predictive model on sales

3.1 Time Series Feature Extraction Plan

Since our goal is to predict sales of next month, the first two obvious features that are useful in prediction are sales in previous months and rolling average of monthly sales.

Then, by visualizing time series data in Power BI, drawing a line chart of sales changes over time (see plot below), we can clearly find that sales have seasonality, and there will be same sales trends in the same month of each year. Therefore, the time information of the sales to be predicted should be considered. Hence, year, month and quarter of the sales to be predicted are included in our feature extraction plan. (the last three columns).





To sum up, we have considered a total of five time series features, which are the sales of the last month, the rolling average sales of the previous three months, and the month, quarter, and year information of the sales to be predicted. (See the figure below). The 'Total Sales' column is the actual sales of the given month in column 'Months', and it's used as the actual label of the data frame. (Requirement 6)

	Months	Brand	Total Sales	Previous Month Sales	Rolling Average 3M	year	month	quarter
1473	2018-12-01	1964 Supply Co.	11862.458300	5402.87306	14830.434353	2018	12	4
2004	2019-01-01	1964 Supply Co.	3999.035200	11862.45830	10292.848487	2019	1	1
2540	2019-02-01	1964 Supply Co.	2417.479970	3999.03520	7088.122187	2019	2	1
3096	2019-03-01	1964 Supply Co.	1607.563310	2417.47997	6092.991157	2019	3	1
3664	2019-04-01	1964 Supply Co.	292.135879	1607.56331	2674.692827	2019	4	2

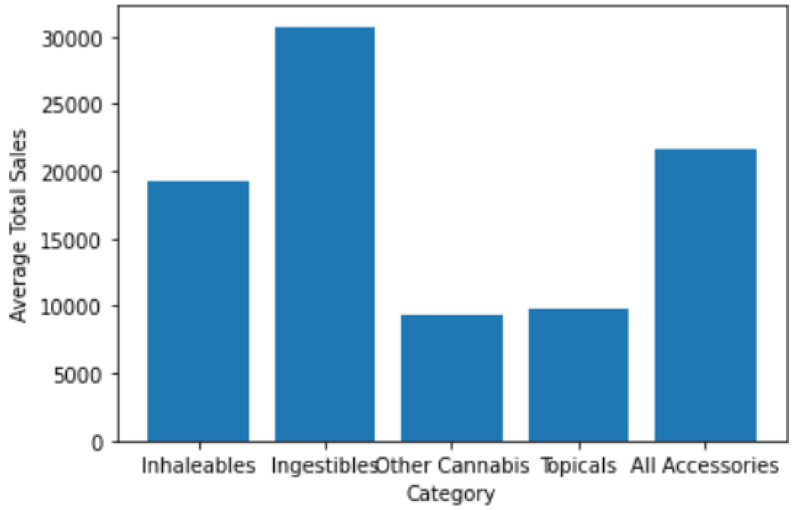
3.2 Data Strategy

Provide an explanation or justification for why you chose the data you did, and also detail any experiments you ran and the results.

We have already discussed the reason why we take these five timeseries features into account. Besides, we also introduced brand-level features, that is, the category to which the brand belongs. There are five categories among brands. A brand may contain multiple types of products. The table below shows the statistics of the number of categories.

Category	Count
Inhaleables	121859
Ingestibles	15554
Other Cannabis	3074
Topicals	2567
All Accessories	1923

Moreover, we compared the total sales for each category (As shown in the figure below).



It is found that the average total sales of various categories of products have differences to a certain extent, which means that there is a high probability that the category will have some impact on the total sales. Therefore, we take the category information into consideration, extracting it as a feature. Because it is a categorical feature, we should one-hot encode it. The data frame that includes timeseries features and brand-level features is shown below. After this, we one-hot encode the year, month and quarter, so that the model will train different parameters for different months that can **capture seasonality.**

	Months	Brand	Total Sales	Previous Month Sales	Rolling Average 3M	Inhaleables	Topicals	Ingestibles	All Accessories	Other Cannabis	year	month	quarter
1473	2018-12-01	1964 Supply Co.	11862.458300	5402.87306	14830.434353	0	0	0	0	0	2018	12	4
2004	2019-01-01	1964 Supply Co.	3999.035200	11862.45830	10292.848487	0	0	0	0	0	2019	1	1
2540	2019-02-01	1964 Supply Co.	2417.479970	3999.03520	7088.122187	0	0	0	0	0	2019	2	1
3096	2019-03-01	1964 Supply Co.	1607.563310	2417.47997	6092.991157	0	0	0	0	0	2019	3	1
3664	2019-04-01	1964 Supply Co.	292.135879	1607.56331	2674.692827	0	0	0	0	0	2019	4	2

3.3 Employ an ensemble method to predictive model exercise

First, we tried a basic Linear Regression predictive model to predict sales, the code is shown below. (Requirement 7)

```
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression(normalize=True)
lin_reg.fit(X_train_prepared, y_train)
coeff_df = pd.DataFrame(lin_reg.coef_, full_pipeline.get_feature_names_out(), columns=['Coefficient'])
coeff_df.sort_values('Coefficient')
```

In addition to the previously trained single linear regression model, we also implemented an ensemble method——Random Forest. The code is shown below. This kind of prediction model have following benefits:

1) Able to run efficiently on large data sets
2) Introduced randomness, not easy to overfit
3) Random forest has good anti-noise ability.
4) Can handle very high dimensional data

```
forest_reg_optimize = RandomForestRegressor(n_estimators=40 ,max_depth=15 ,oob_score=True ,n_jobs=-1, random_state=42)
forest_reg_optimize.fit(X_train_prepared, y_train)
pred_1 = forest_reg_optimize.predict(X_test)
```

3.4 Cross-Validation

K-Fold Cross Validation is not suitable for parameter tuning with Time Series Data according to the Question @97 on Piazza, since there is a problem of using model trained by future data to predict historical data. So, in this scenario, we just choose all data for September 2021 as the test set. However, in order to comply with the requirement 10 of *Specific Coding Requirements*: we perform a cross-validation on the Random Forest model. We use GridSearchCV method to select optimal model hyperparameters (the code is shown below). GridSearchCV method merges the gridsearch and cross-validation steps, which can select hyperparameters and cross validate model simultaneously. We only need to assign a value to the cv parameter in this method, which determines the k in k-fold cross-validation. By passing 10 to the *cv* parameter, 10-fold cross-validation is employed in our Random Forest model.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV

param_grid = [
    {'n_estimators':[30,40,50,80,100],'max_depth':[5,8,10,12,15,20,25]},
]

forest_reg = RandomForestRegressor(oob_score=True ,n_jobs=-1, random_state=42)
grid_search = GridSearchCV(forest_reg, param_grid, cv=10,
                           scoring='neg_mean_squared_error',
                           return_train_score=True)

grid_search.fit(X_train_prepared,y_train)
```

Part B: Findings on key indicators for successful product

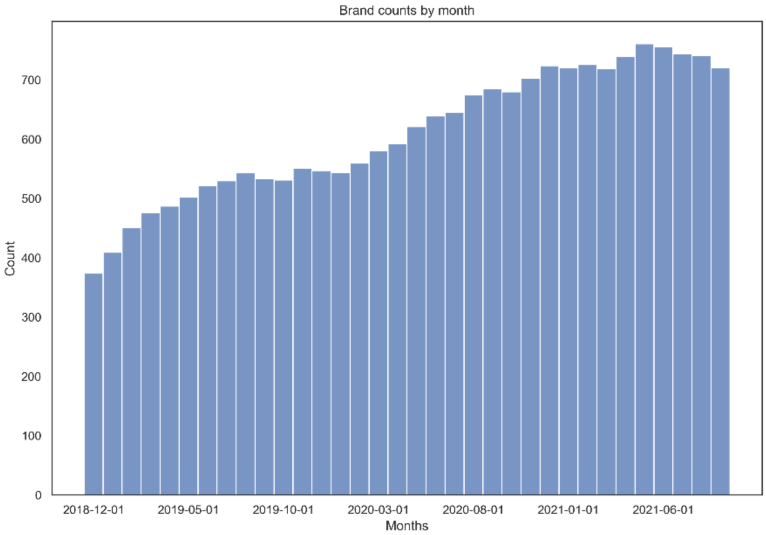
3.5 Why choose to use the data in the BrandDetail table?

The criterion for judging the success of a product should be the gradual increase in monthly sales of the product, or how much the growth rate of it exceeds the growth rate of total market sales. However, we only have sales time series data for TOP50 products. If you use them to study the driving factor of successful products, we will face the following problems:

a) There are many features in *BrandDetail.csv* that account for the success of products, and each feature has many discrete categories. We cannot run a linear regression on a dataset with only 50 samples but with more than 10 features which will cause overfitting.

b) In addition, *Top50ProductsbyTotalSales-Timeseries.csv* contains the data of 50 products from October 2018 to September 2021, which indicates that if some new products appear after October 2018, and have reached a good sales record, they will not be included in the data set. This will lead to a bias in our analysis: only the factors for the success of products that went on the market earlier are studied.

In recent years, the number of brands in the market has increased rapidly according to the figure below, resulting in a low degree of differentiation between brands, and many brands are offering similar products. Moreover, consumers' recognition of the brand also comes from the design and positioning of the product, so we believe that a good brand is determined by the product, not the brand that determines the sales of the product. The product information provided in *Brand Detail.csv* is exactly what we need.



3.6 How do we deal with data and why we should deal with it this way

When analyzing the features of each product, we do some preprocessing on the features:

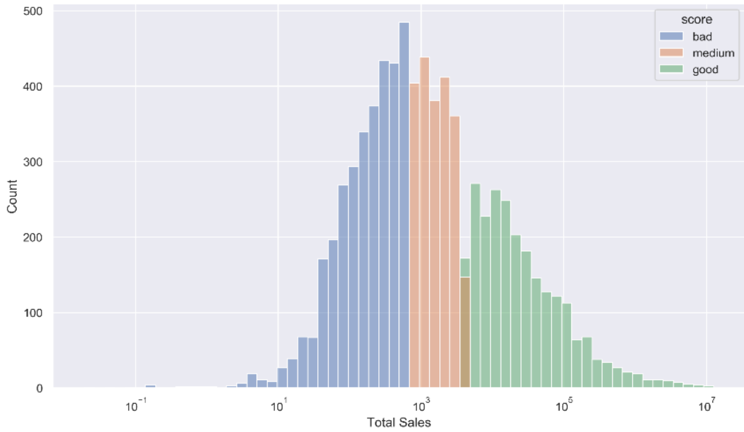
a) We deleted ‘State’ and ‘Channel’ column, because they all take the same value, so they don’t work for the analysis. We also deleted the ‘Brand’ column because we believe that the brand information has been reflected in other features.

b) ARP is not included in the analysis because we have grouped ARP into "ARP Category". The segmentation interval is as follows: [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, inf]

c)

d) **Deal with missing values**: For each feature, we first remove the null value. Because the null value is often due to products that are not suitable for evaluation with this feature, for example, when the value of Category L2 is *Topicals*, the ‘Is Flavored’ feature are all null values, and these lines should not be included when considering about the feature ‘Is Flavored’. Therefore, we should not impute missing values, but just drop them.

e) **Label data by ‘Total sales’:** For the filtered data set, we constructed a new data frame containing the two columns containing the feature to be evaluated and **‘score’**. Among them, the score is the label that divides the dataset into three categories: good, medium, and bad according to the total sales. The sample segmentation results of the entire data set are as follows:



f) **Building metrics:**

1. **Probability of occurrence in three scored groups:** Inspired by the Bayes Theorem, we use the conditional probability of the category x_i occurs given the condition that this product is a successful product (is labeled as *'Good'* in the 'score' column) as the measure of the quality of this category in certain feature. The formula is as follows: $P(X_j = x_i | score =' Good')$. Where X_j is the j^th feature of the dataset and x_i is the i^th possible value of the X_j feature.
2. **Sorted the probability:** After finding the conditional probability of values given their group belonging for each feature, we then sort the probability of values in the feature X_j from high to low. In other words, in the category of 'Good', the value of the feature with the highest number of occurrences will be ranked first. In the category 'Good', the second most frequent one came in second.
3. **Out performance indicator:** We also built an indicator called *out performance*. This indicator is used to measure how much can the value of a certain feature can boost sales. It is calculated as follow:

$$out\ performance = \frac{P\left(X_j = x_i \middle| score =' Good'\right) - P\left(X_j = x_i \middle| score =' Good'\right)}{\frac{1}{2}\left(P\left(X_j = x_i \middle| score =' Good'\right) + P\left(X_j = x_i \middle| score =' Good'\right)\right)}$$

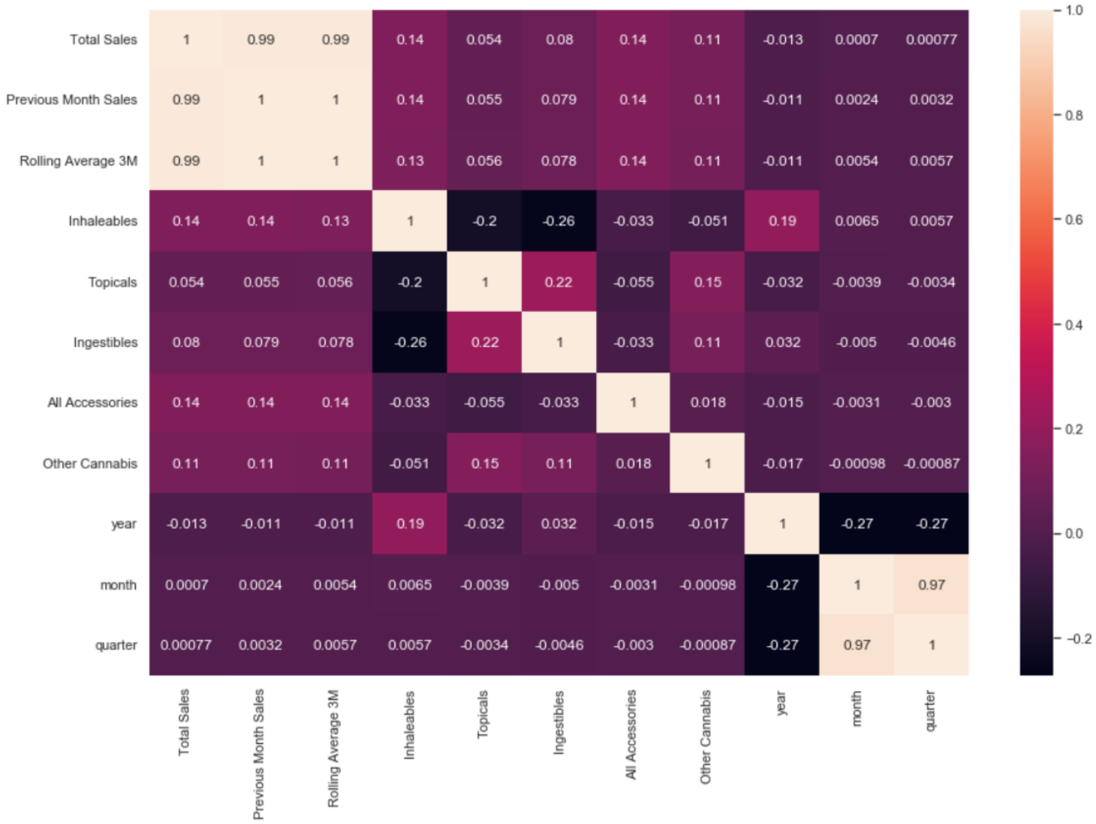
3.7 Output the metrics

Finally, we construct a data frame and put the four recommended value categories of each feature (ranked by the probability of occurrences in 'Good' group) into the column named as this feature. Output to a data frame called *feature_recommendation_df*. And output their corresponding out performance coefficients as a data frame called *feature_recommendation_score_df*. We also output a data frame with the probability of occurrences for 4 recommended values in each feature to show the empirical market share that achieved in current market. Results are shown in the *Result Chapter.*

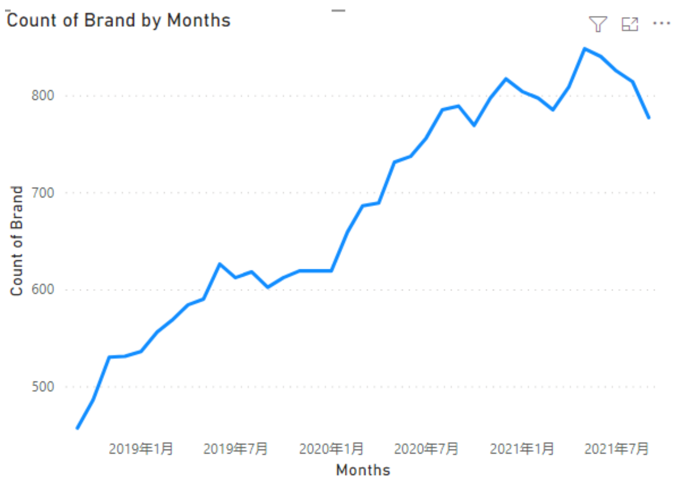
4 Results

4.1 Basic statistics on variables

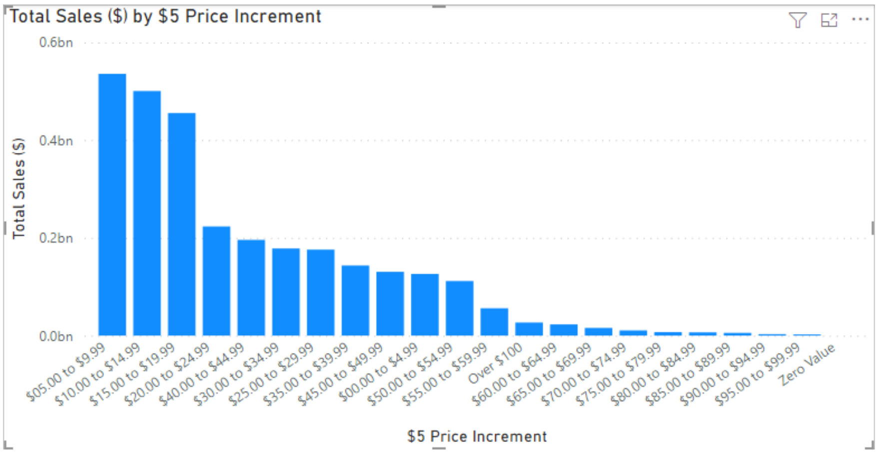
First, let's discuss the findings from the correlation matrix below. We can see that the sales of last month and the rolling average sales of the previous three months are highly correlated with the label-Total Sales. From another perspective, the correlation coefficients between the features are relatively small, indicating that there is basically no collinearity between them, which is good for regression prediction problems



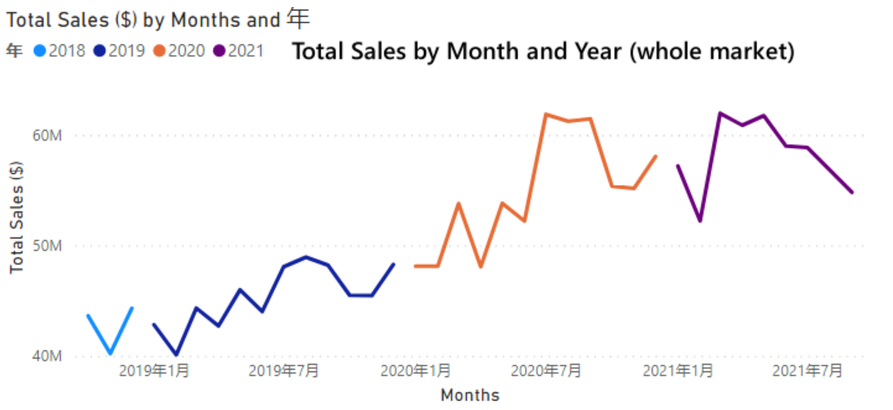
We also run some basic statistics on the variables, results are shown below:



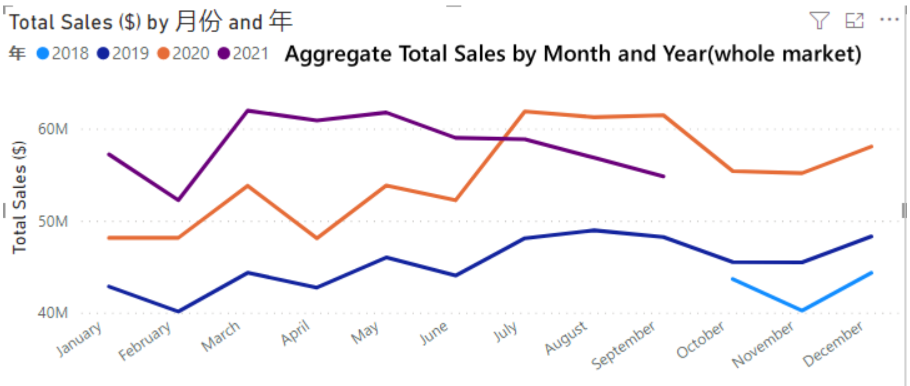
The number of brands is rising rapidly.



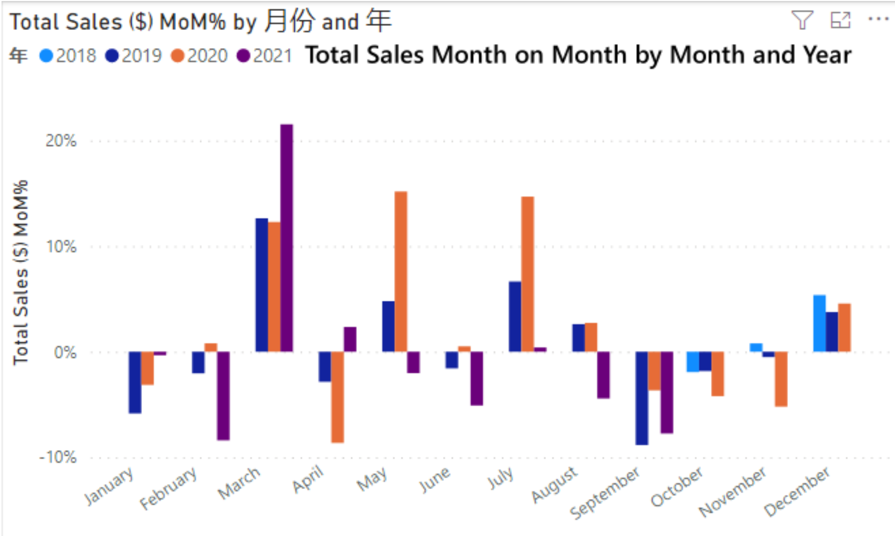
Majority of the sales falls in the low-price categories.



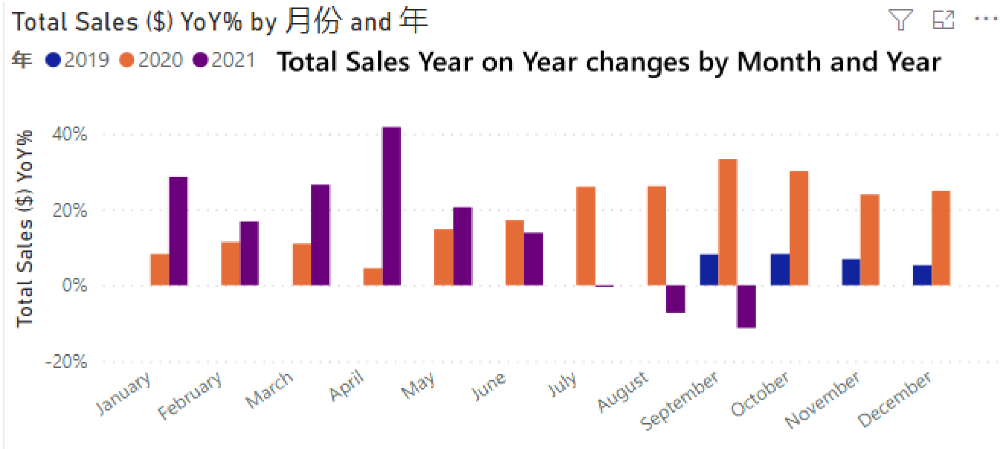
Total Sales of the whole market rises from 40 million to around 55 million per month.



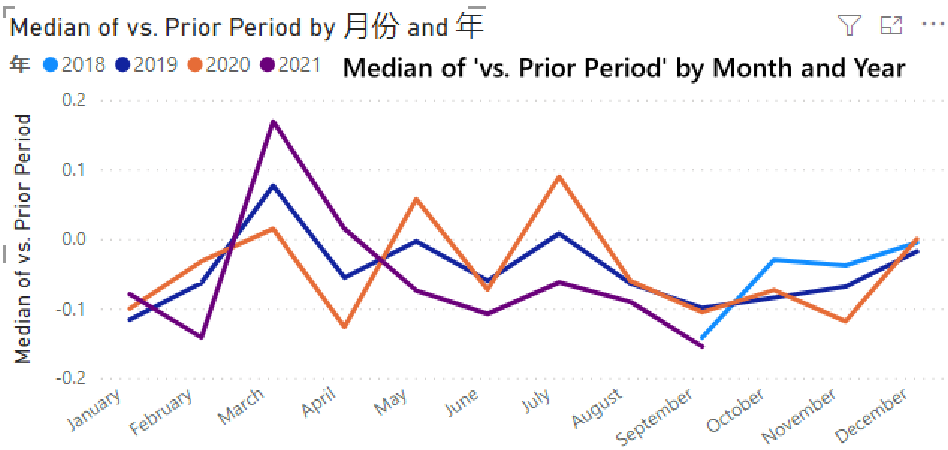
Total sales has obvious monthly seasonality, such as all going up from February to March and going down from May to June.



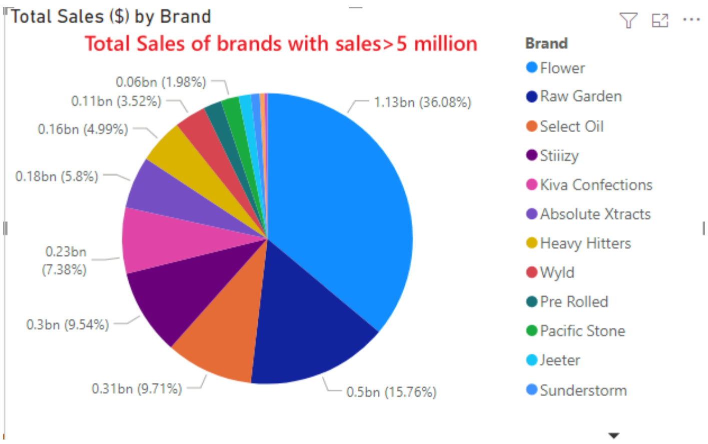
The changes of total sales of whole market Month-on-Month shows the seasonality.

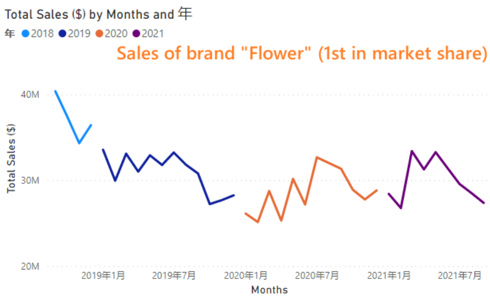


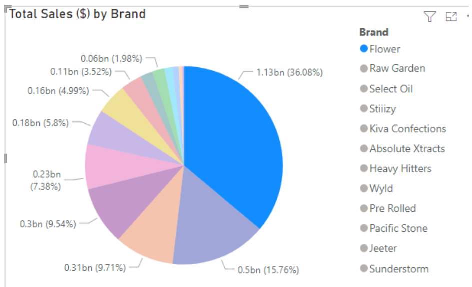
Comparing the total sales Year-on-Year changes, we can see the strong growth in 2020 and the slowing down in mid-2021.

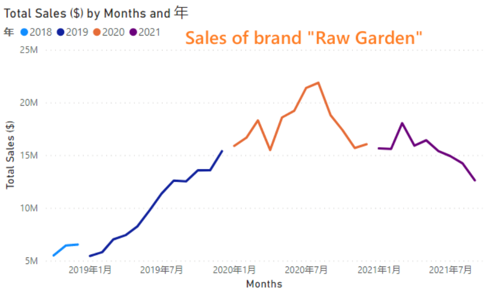


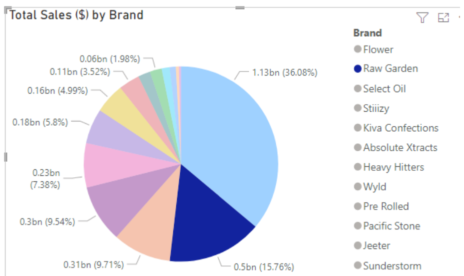
The fluctuation of the median of feature “vs. Prior Period” also shows strong monthly seasonality.



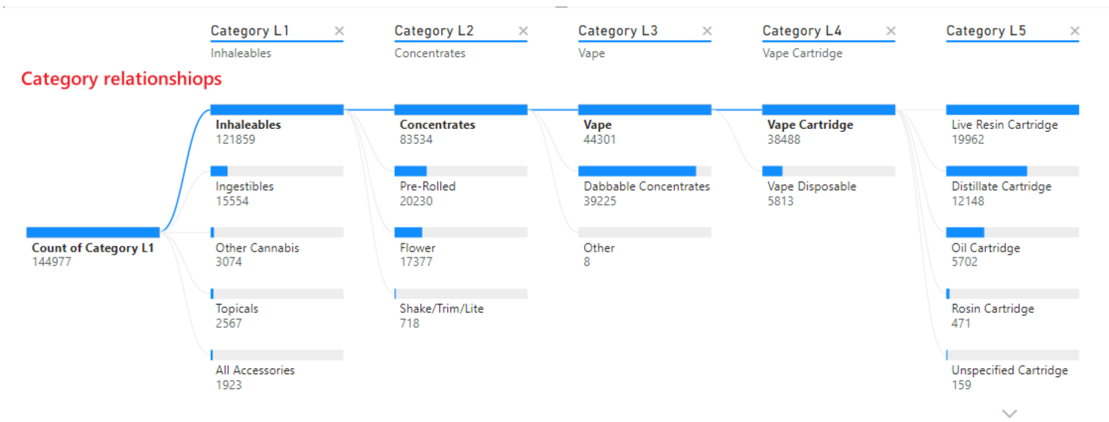




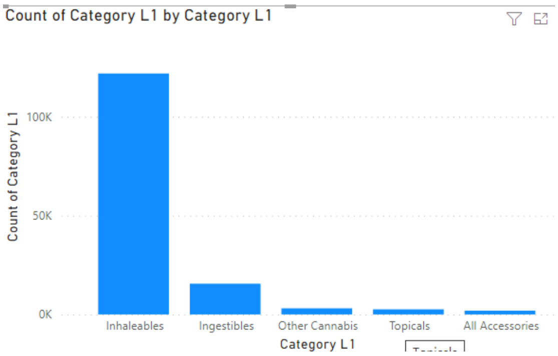


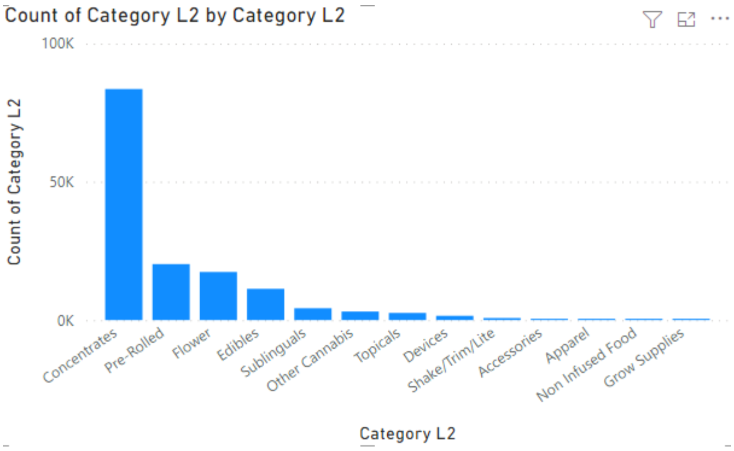


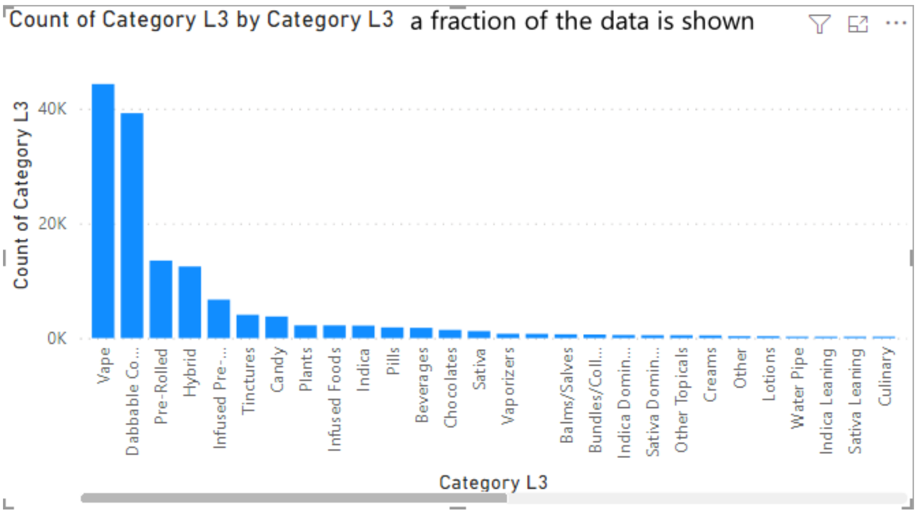
Looking at the cumulative total sales group by brands, we can found that this is a emerging market with a lot of brands competing with each other. And the growth rates of sales-leading brands are slowing down.



There are a lot of categories for classify the product.


Count of Category L1 by Category L1
Count of Category L1
100K
50K
0K
Inhaleables
Ingestibles
Other Cannabis
Topicals
All Accessories
Category L1


Count of Category L2 by Category L2
Count of Category L2
100K
50K
0K
Concentrates
Pre-Rolled
Flower
Edibles
Sublinguals
Other Cannabis
Topicals
Devices
Shake/Trim/Lite
Accessories
Apparel
Non Infused Food
Grow Supplies
Category L2


Count of Category L3 by Category L3
a fraction of the data is shown
Count of Category L3
40K
20K
0K
Vape
Dabbable Co...
Pre-Rolled
Hybrid
Infused Pre-...
Tinctures
Candy
Plants
Infused Foods
Indica
Pills
Beverages
Chocolates
Sativa
Vaporizers
Balms/Salves
Bundles/Coll...
Indica Domin...
Sativa Domin...
Other Topicals
Creams
Other
Lotions
Water Pipe
Indica Leaning
Sativa Leaning
Culinary
Category L3

4.2 Result of Linear Regression model

After data pipelining, we use Statsmodels package, and OLS（ordinary least square）analysis is implemented. The result of OLS Regression and statistical metrics is shown as the table below. It can be found the P-value of most of features in t-statistics are less than 0.05, which indicates that there are statistically significant relationships between features and the label. For example, the sales of last month, the rolling average sales of the previous three months, and category the brand includes. These features are important to fit the model. (Requirement 7 p-values test)

```
==============================================================================
Dep. Variable:            Total Sales   R-squared:                       0.987
Model:                            OLS   Adj. R-squared:                  0.987
Method:                 Least Squares   F-statistic:                 7.311e+04
Date:                Sat, 27 Nov 2021   Prob (F-statistic):               0.00
Time:                        16:02:40   Log-Likelihood:            -2.8175e+05
No. Observations:               20734   AIC:                         5.635e+05
Df Residuals:                   20712   BIC:                         5.637e+05
Df Model:                          21
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1           1.41e+06   1.64e+04     86.242      0.000    1.38e+06    1.44e+06
x2          2.488e+05   1.63e+04     15.224      0.000    2.17e+05    2.81e+05
x3          6.932e+04   1603.683     43.226      0.000    6.62e+04    7.25e+04
x4          8.267e+04   1740.557     47.497      0.000    7.93e+04    8.61e+04
x5          7.882e+04   2325.198     33.897      0.000    7.43e+04    8.34e+04
x6          7.318e+04   2842.878     25.740      0.000    6.76e+04    7.87e+04
x7          7.103e+04   1714.926     41.416      0.000    6.77e+04    7.44e+04
x8          8.097e+04   1783.781     45.390      0.000    7.75e+04    8.45e+04
x9          7.194e+04   1806.580     39.823      0.000    6.84e+04    7.55e+04
x10         8.005e+04   2240.922     35.721      0.000    7.57e+04    8.44e+04
x11         7.062e+04   2684.950     26.301      0.000    6.54e+04    7.59e+04
x12         8.138e+04   3418.406     23.805      0.000    7.47e+04    8.81e+04
x13         5.342e+04   8784.416      6.081      0.000    3.62e+04    7.06e+04
x14         3.497e+04   3280.663     10.661      0.000    2.85e+04    4.14e+04
x15         4.012e+04   3165.740     12.674      0.000    3.39e+04    4.63e+04
x16         2.348e+04   3542.706      6.628      0.000    1.65e+04    3.04e+04
x17        -8365.9803   3883.531     -2.154      0.031    -1.6e+04    -753.955
x18        -1.271e+04   3832.410     -3.317      0.001   -2.02e+04   -5201.768
x19          6.69e+04   3806.063     17.577      0.000    5.94e+04    7.44e+04
x20         3940.7154   3731.696      1.056      0.291   -3373.703    1.13e+04
x21         3.428e+04   3681.591      9.310      0.000    2.71e+04    4.15e+04
x22          863.7327   3659.878      0.236      0.813   -6309.915    8037.381
x23         4.196e+04   3639.199     11.530      0.000    3.48e+04    4.91e+04
x24          1.39e+04   3613.832      3.846      0.000    6816.778     2.1e+04
x25        -1.723e+04   3624.677     -4.754      0.000   -2.43e+04   -1.01e+04
x26        -1968.5633   4575.843     -0.430      0.667   -1.09e+04    7000.448
x27         -510.2848   4522.908     -0.113      0.910   -9375.540    8354.971
x28         3.094e+04   4458.877      6.940      0.000    2.22e+04    3.97e+04
x29         4.582e+04   1831.848     25.012      0.000    4.22e+04    4.94e+04
x30         3.908e+04   1786.010     21.882      0.000    3.56e+04    4.26e+04
x31         3.863e+04   1754.639     22.015      0.000    3.52e+04    4.21e+04
x32         2.846e+04   2084.407     13.656      0.000    2.44e+04    3.26e+04
==============================================================================
Omnibus:                    23045.517   Durbin-Watson:                   2.103
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         33543629.963
Skew:                           4.724   Prob(JB):                         0.00
Kurtosis:                     199.820   Cond. No.                     2.29e+16
==============================================================================
```

The performance of linear regression model is measured by the following metrics:

explained_variance	0.9929
r^2	0.9929
MAE	62062.7429
MSE	17195648628.1742
RMSE	131132.18

The value of R^2 is close to 1, however the value of MSE is large. In order to figure out what is the problem and see how our model performs over data with various sales values, three charts with different actual sales range are plotted as follows.

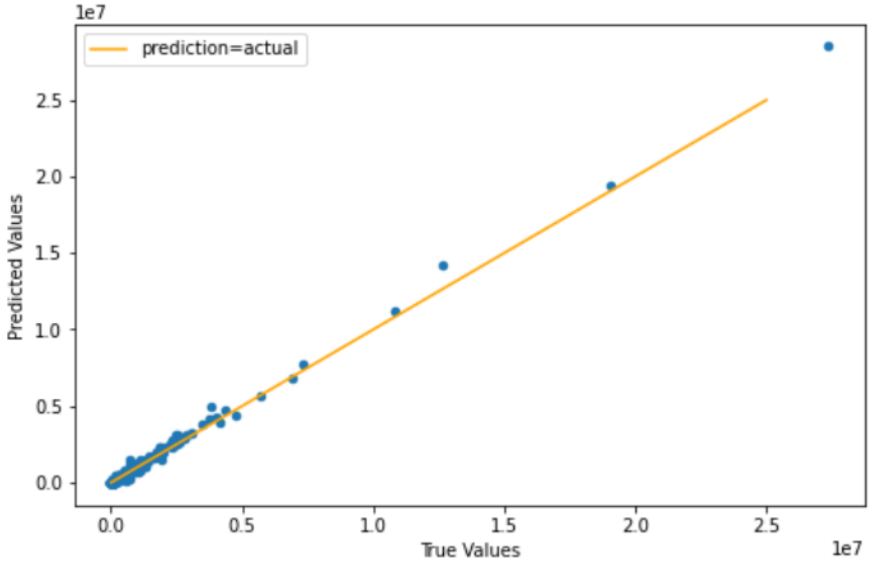


Figure 1 all True/Predicted values (no range limit)

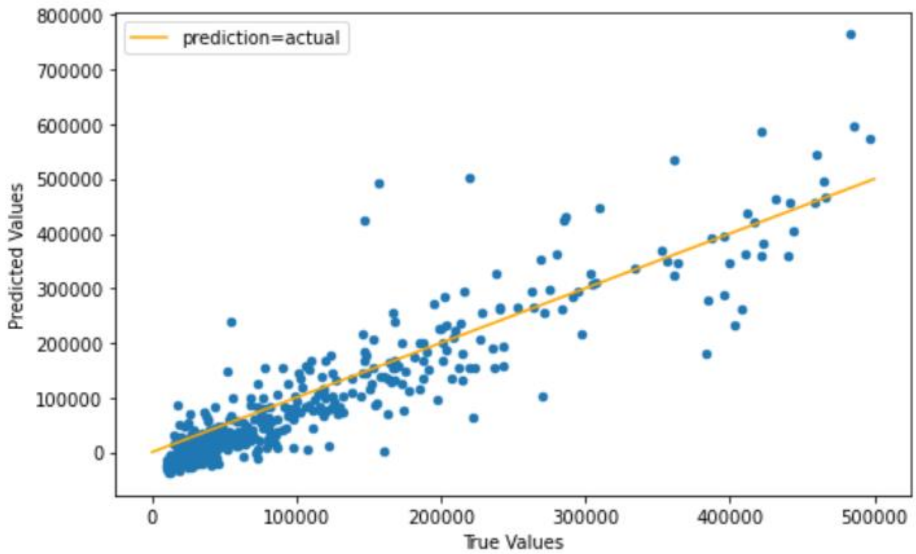


Figure 2 True/Predicted values between 10,000 and 500,000

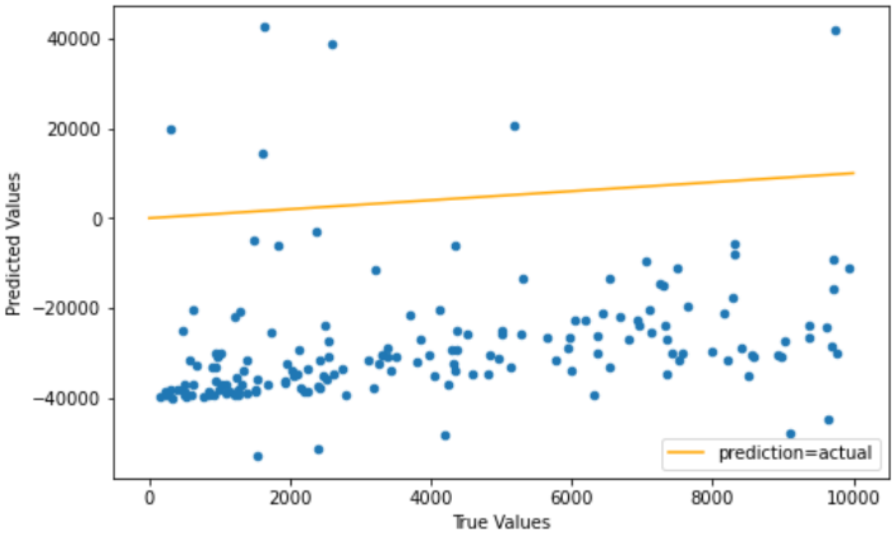


Figure 3 True/Predicted values less than 10,000

Figure 1 is a scatter plot containing all predicted and true values. Figure 2 is a scatter plot only including scatter plots where the true sales is greater than 10,000 and less than 500,000 and the corresponding predicted sales. Figure 3 is a scatter plot only including scatter plots where the true sales is less than 10,000 and the corresponding predicted sales.

It can be found that the model performs well in forecasting large sales (larger than 10000), but the prediction performance is bad for small sales (less than 10000). This may also cause by the sensitivity to outliers of linear regression model. At this point we can know the reason why there is a good value of R^2, but a large MSE value. It is because the sales of most brands are less than 10,000. In order to minimize the loss function as much as possible, the model will pay more attention to the prediction of brands with large sales but ignore the brands with small sales, and the prediction performance of these bands with small sales is not good. This can also be viewed as the linear regression model's sensitivity to outliers.

Therefore, there are two solutions:

1) build models specifically for brands with sales under 10,000;

2) build other models on the whole dataset and check its performance on data with sales less than 10,000.

Chapter 4.3 will show the result of second solution.

4.3 Employ an ensemble method-Random Forest model (Requirement 9)

The performance of random forest model is measured by the following metrics:

explained_variance	0.9922
r^2	0.9919
MAE	56387.0258
MSE	19604199813.1873
RMSE	140014.9985

The value of R^2 is close to 1, however the value of MSE is still large. The same problem happens again.

We then ran the cross-validation process on the Random Forest model to find out the beset parameters:
Again, GridSearch method is used to optimize the Random Forest model. The parameters we want to optimize are the number of trees in the forest (n_estimators) and the maximum depth of the tree (max_depth). A range of choices are provided as below.
'n_estimators':[30,40,50,80,100], 'max_depth':[5,8,10,12,15,20,25]
The method finally selected n_estimators = 40, max_depth=15 as the optimal parameters.

```
In  [38]: from sklearn.ensemble import RandomForestRegressor
          from sklearn.model_selection import GridSearchCV

          param_grid = [
              {'n_estimators':[30, 40, 50, 80, 100],'max_depth':[5, 8, 10, 12, 15, 20, 25]},
          ]

          forest_reg = RandomForestRegressor(oob_score=True ,n_jobs=-1, random_state=42)
          grid_search = GridSearchCV(forest_reg, param_grid, cv=10,
                                     scoring='neg_mean_squared_error',
                                     return_train_score=True)

          grid_search.fit(X_train_prepared,y_train)

Out[38]: GridSearchCV(cv=10,
                      estimator=RandomForestRegressor(n_jobs=-1, oob_score=True,
                                                      random_state=42),
                      param_grid=[{'max_depth': [5, 8, 10, 12, 15, 20, 25],
                                   'n_estimators': [30, 40, 50, 80, 100]}],
                      return_train_score=True, scoring='neg_mean_squared_error')
```

```
In  [39]: grid_search.best_estimator_

Out[39]: RandomForestRegressor(max_depth=15, n_estimators=40, n_jobs=-1, oob_score=True,
                               random_state=42)
```

After find out the best set of parameters using cross validation, we then plot three charts like above with different range of actual sales. As can be seen from the Figure 6, the performance of the random forest model is better than the linear regression model in the prediction of small sales.

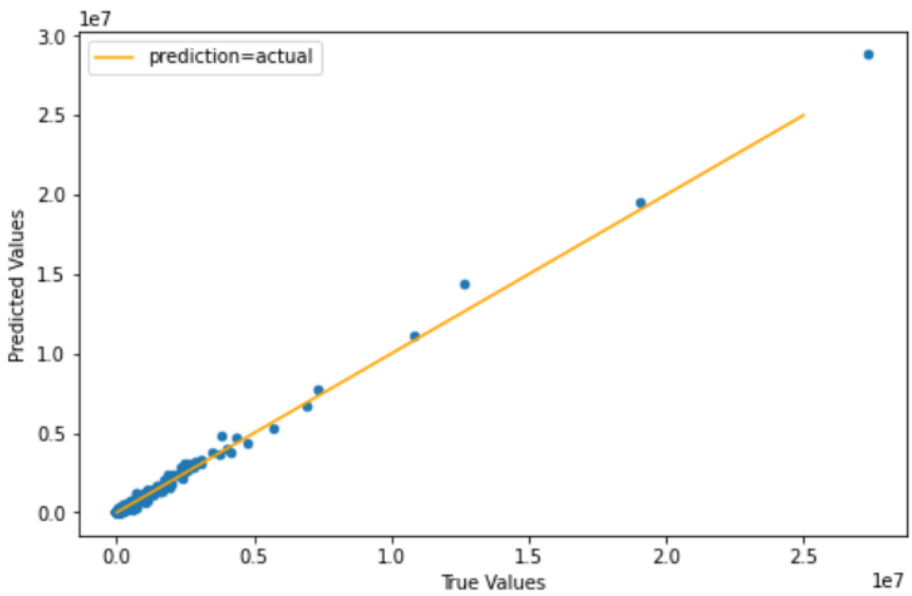


Figure 4 all True/Predicted values (no range limit) on RF model

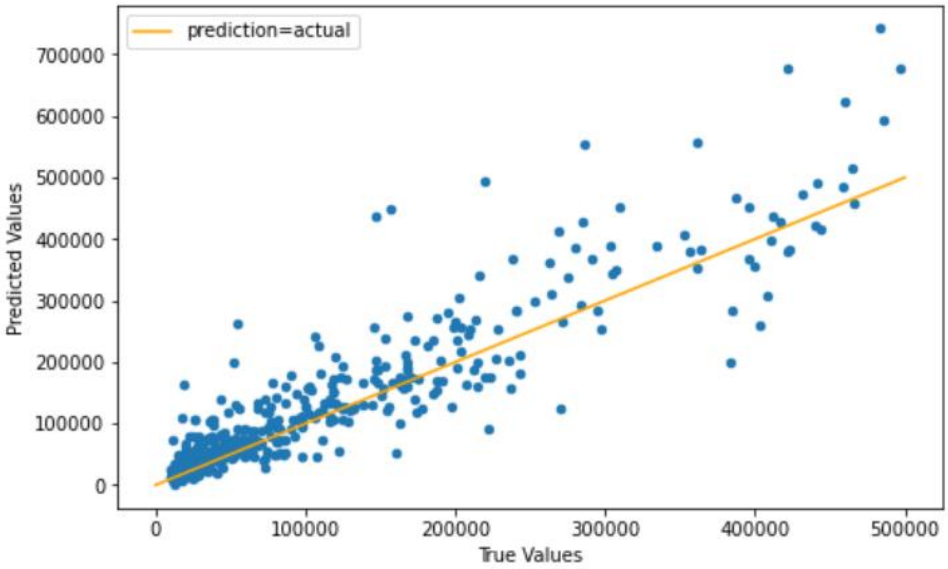


Figure 5 True/Predicted values between 10,000 and 500,000

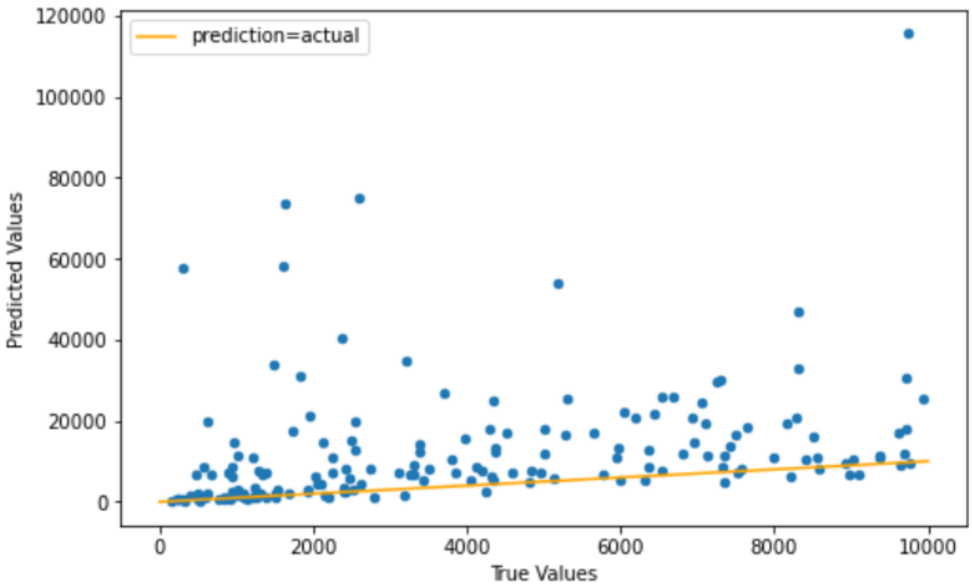


Figure 6 True/Predicted values less than 10,000

From figure 4-6, we can see that Random Forest model has greatly improved the accuracy for total monthly sales less than 10,000 while still keeping great accuracy for samples with larger sales. Overall, our solution 2 (using other models) works! The performance of RF model is better than the basic linear regression model.

4.4 Experiment with custom models and comparison of all models

Since a single regression model does not have a good predictive performance on brands with small sales, we are now considering the **solution 1):** making multiple regression predictions by segmenting dataset based on sales and train three separate models on them. As shown earlier, the model does not work well when predicting brands with sales less than 10,000, so we choose 10,000 as the threshold and divide the original dataset into two. The first part includes those data whose sales of last month is less than 10,000, the second part includes data whose sales of last month is larger than 10,000. Note that we can only separate data by features in training set, which means this separation will be different from our previous separation using the 'True values', i.e the actual sales of a certain brand in the given month.

4.4.1 Separated Linear Regression Model

First, we try to implement linear regression model for both datasets. The metrics are shown as below:

First dataset (sales of last month < 10000)	
r^2	-150.8929
RMSE	86573.3306
Second dataset (sales of last month > 10000)	
r^2	0.992
RMSE	154044.8618

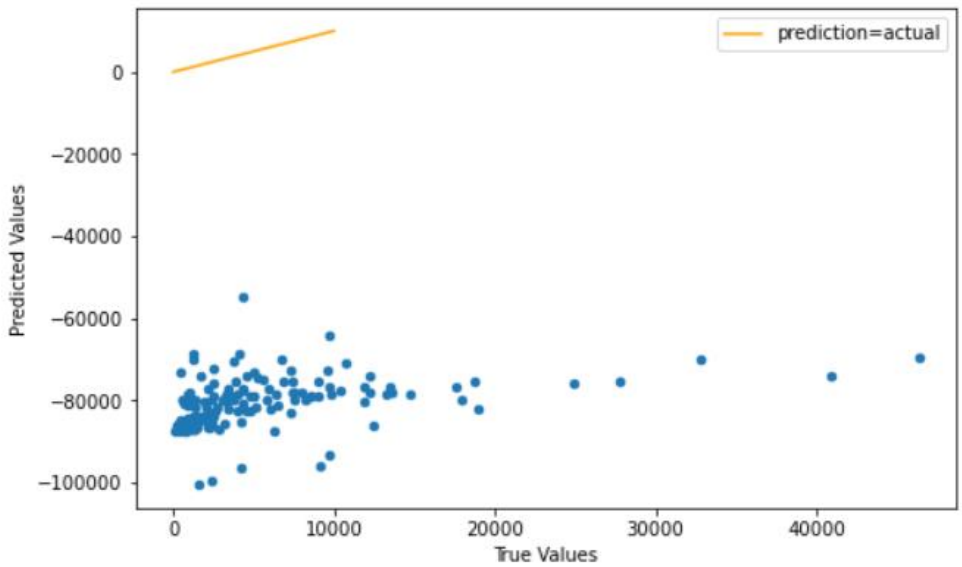


Figure 7 True/Predicted values with last month sales less than 10,000 using separated linear regression model

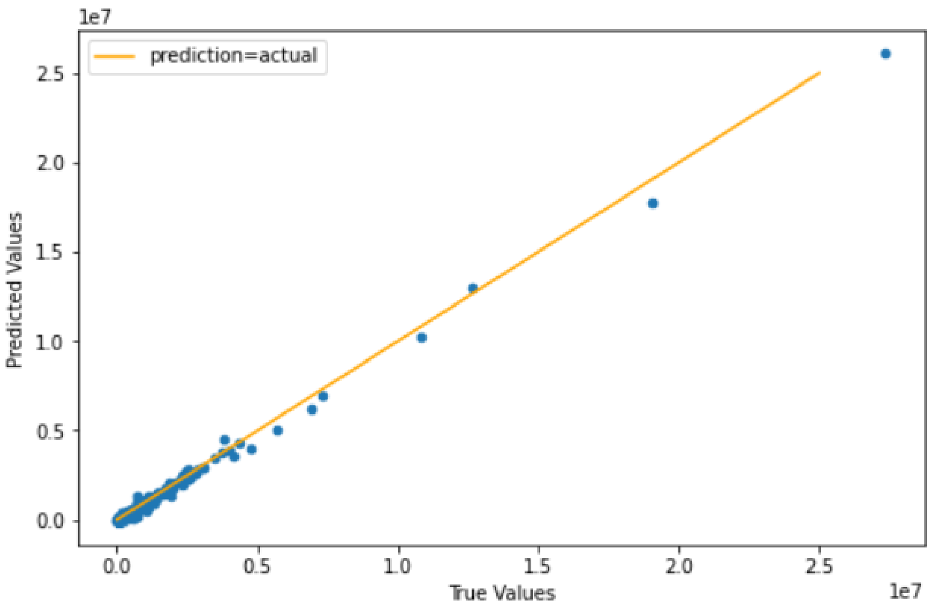


Figure 8. True/Predicted values with last month sales greater than 10,000 using separated linear regression model

We can see that the linear regression model performs well on the second dataset, but not on the first dataset. Therefore, we only need to experiment other models for the first dataset (finding better models on divided dataset, combing solution 1 and 2). So our next step is to find a model that deliver better performance on the 'small brand' dataset (actual monthly sales under 10,000).

4.4.2 Separated Random Forest model and XGBoost model

Now we are going to apply random forest model and XGBoost model for the first dataset.
The forecast results of random forest model are shown below.

First dataset (sales of last month<10000)	
r2	0.0733
RMSE	6762.0227

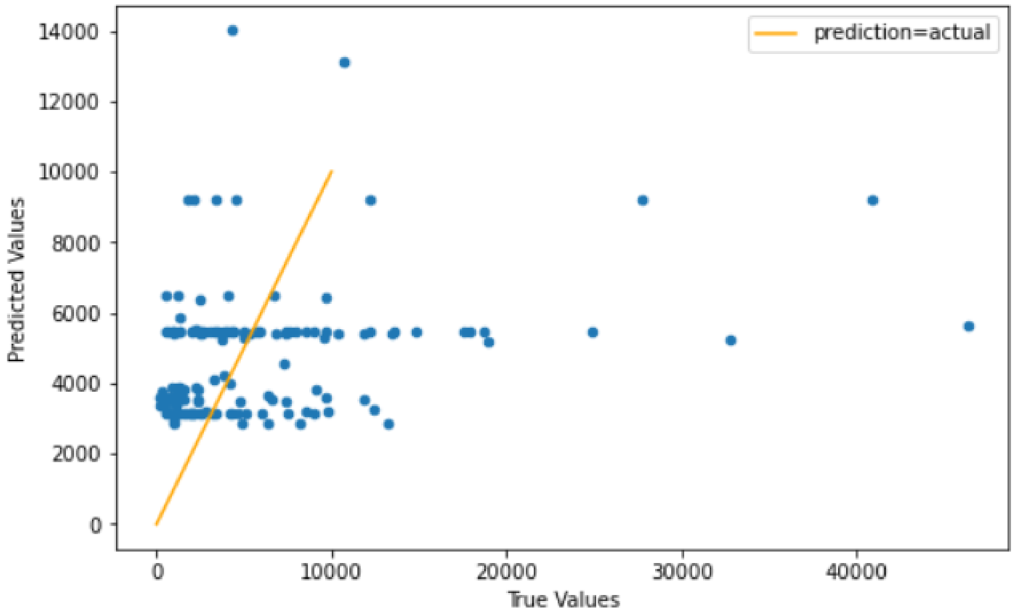


Figure 9. True/Predicted values with last month sales greater than 10,000 using separated Random Forest model

The forecast results of XGBoost model are shown below.

First dataset (sales of last month<10000)	
r2	-0.0379
RMSE	7156.484

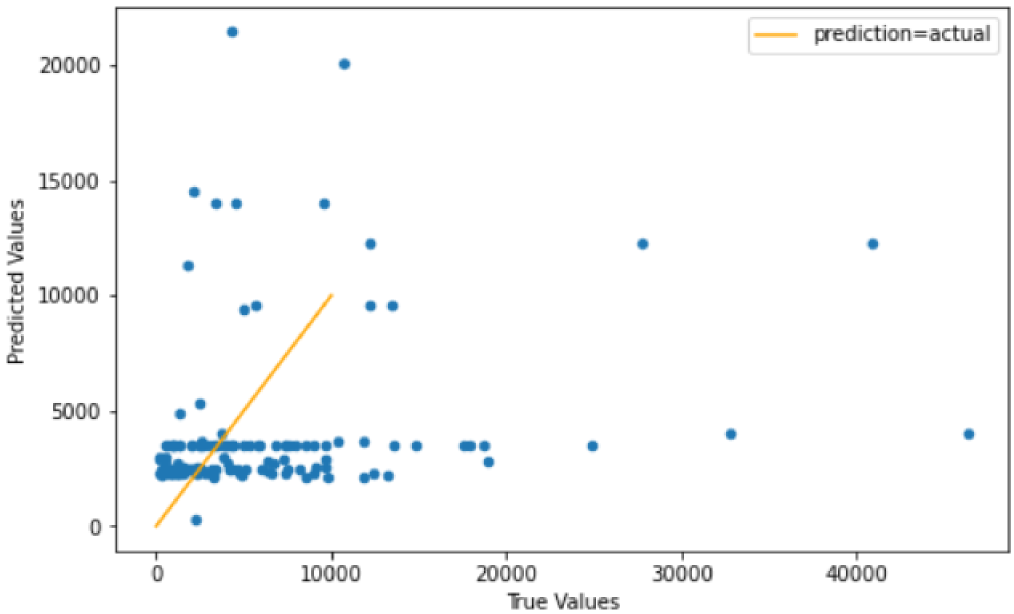


Figure 10. True/Predicted values with last month sales greater than 10,000 using separated XGBoost model

From figure 9 and figure 10, we can see that we may have the same predictive values for various true values, which indicates that our model can only output a few number of predictions given there are a lot of different true values. In other words, our models haven't learned enough information to give various predictions. That's a symptom of underfitting.

In summary, we tried 3 models (Linear Regression, Random Forest, XGBoost) for the first dataset, however, the segmented regression cannot improve the prediction performance. The reason may be that brands with small sales are unstable in the market, the selected features are not enough to capture the trend of those brands on sales.

4.4.3 XGBoost model on overall dataset

Therefore, in order to give more information for our model to better train themselves, we use our full dataset to train the XGBoost model. Another reason why we using XGBoost is that we can see from figure 7 that the predictions of our model have a 'bias' to the actual values. We may be able to make better predictions by adding a bias value to original predictions.

XGBoost is a model that learns the differencs between our predicted values and actual values. Therefore it's a great fit for this task.

Finally, XGBoost model is employed on the whole dataset. The result are shown as below.

explained_variance	0.9941
r2	0.9941
MAE	48363.1028
MSE	14175169806.2314
RMSE	119059.5221

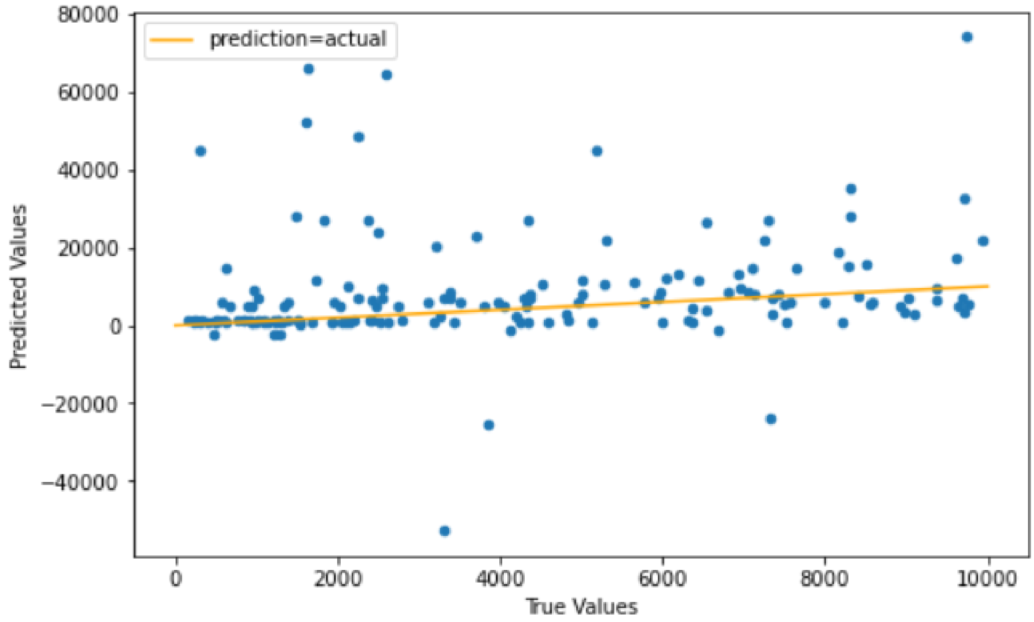


Figure 11. True/Predicted values with last month sales less than 10,000 using XGBoost model trained on whole dataset

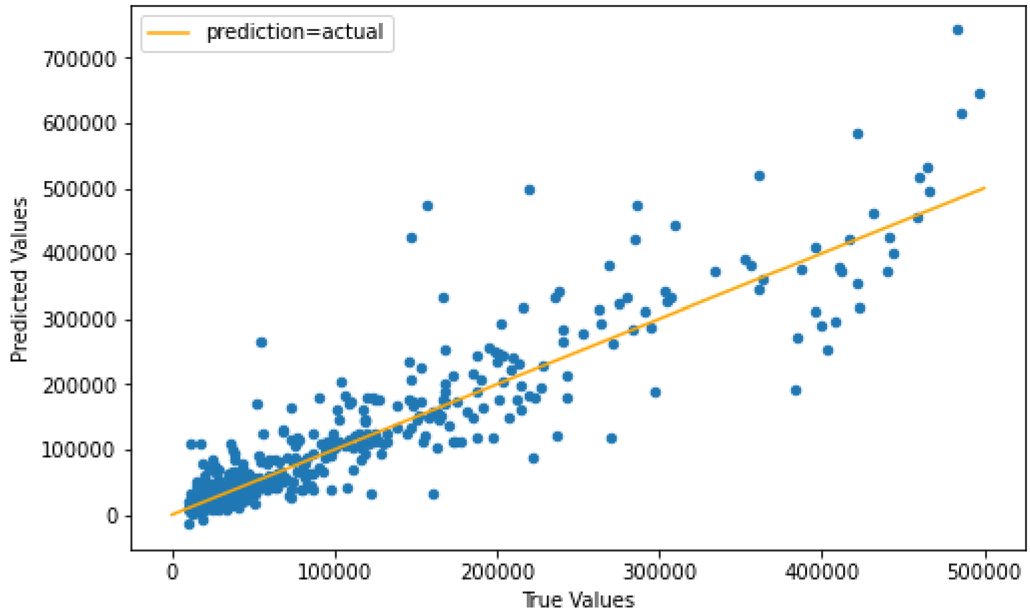


Figure 12. True/Predicted values with last month sales greater than 10,000 using XGBoost model trained on whole dataset

From figure 11 we can see that XGBoost model give considerably well predictions on majority of samples. However, this model also makes ridiculous predictions for some small actual sales value. What's more, it gives a various of different prediction values which indicates that this model has learned more information than the previous XGBoost model trained by only a fraction of a dataset Figure 10).

The performance comparison of the tried models is shown in the table below, the highest performing model is XGBoost model.

	Linear Regression	Random Forest	XGBosst
explained_variance	0.9929	0.9922	0.9941
r2	0.9929	0.9919	0.9941
MAE	62062.7429	56387.0258	48363.1028
MSE	17195648628.1742	19604199813.1873	14175169806.2314
RMSE	131132.18	140014.9985	119059.5221

4.5 Part B: Findings on key indicators for the likely success of a new product launch in the current market

Let's look at sample bar plots got from the part B of Chapter 3:

Figure 13. Top 4 values in ARP category that makes a product success

	Good	Medium	Bad	Average odds	Out performance
5-10	0.116730	0.056844	0.085090	0.0625	0.313545
10-15	0.113977	0.062121	0.101500	0.0625	0.115817
30-35	0.097754	0.110439	0.092829	0.0625	0.051683
15-20	0.095809	0.088215	0.124759	0.0625	-0.262501

The result data frame res_df

As mentioned in Chapter 3, the value 0.116730 in 'Good' column and '5-10' row is the probability of occurrences of '5-10' given that it's a 'Good' product (ranked top 1/3 by cumulative total sales). The 'Average odds' here show the average odd of those conditional probability if those categorical values are evenly distributed in a given group. Here, 'ARP category' has 16 categories, so the average odds $= \frac{1}{16} = 0.0625$. The 'out performance' bar shows the positive impact brought by adapting the \$5-\$10 category in the 'Good' category. It's calculated by:

$$out\ performance = \frac{P(X_j = x_i | score =' Good') - P(X_j = x_i | score =' Good')}{\frac{1}{2}\left(P(X_j = x_i | score =' Good') + P(X_j = x_i | score =' Good')\right)}$$

Where X_j is the j[th] feature of the dataset and x_i is the i[th] possible value of the X_j feature.

The figure 13. shows that the price range of \$5-\$10 is the most popular category in the 'Good' group, appearing in about 11.6% of products labeled as 'Good'. Therefore, we are recommending the \$5-\$10 price category for feature *ARP Category*. Combing the top 4 recommendations for all of our features, we got a new data frame named *feature_recommendation_df*. It's shown below:

Category L1	Category L2	Category L3	Category L4	Category L5	Flavor	Items Per Pack	Item Weight	Total THC	Total CBD	Contains CBD	Pax Filter	Strain	Is Flavored	Mood Effect	Generic Vendor	Generic Items	$5 Price Increment	ARP category
Inhaleables	Concentrates	Vape	Vape Cartridge	Live Resin Cartridge	Watermelon	0	1000mg	0	0	THC Only	Not Pax	Hybrid Strain Blends	Flavored	Not Mood Specific	Non-Generic Vendors	Non-Generic Items	$05.00 to $9.99	5-10
Ingestibles	Flower	Dabbable Concentrates	Live Resin	Distillate Cartridge	Strawberry	1	500mg	100	100	Contains CBD	Pax	Wedding Cake	Not Flavored	Mood Specific	Generic Vendors	Generic Items	$10.00 to $14.99	10-15
Other Cannabis	Pre-Rolled	Hybrid	Vape Disposable	Oil Cartridge	Peanut Butter	10	0.5	1,000	300			Indica Strain Blends					$30.00 to $34.99	30-35
Topicals	Edibles	Pre-Rolled	Rosin	Live Resin Disposable	Dark Chocolate	5	1	10	50			Sativa Strain Blends					$15.00 to $19.99	15-20

By selecting the column of 'Good' in the *res_df* data frame for every feature, we can construct a data frame named *feature_recommendation_score_df* that shows the overall market share taken by each category. The heat map of this data frame is shown below:

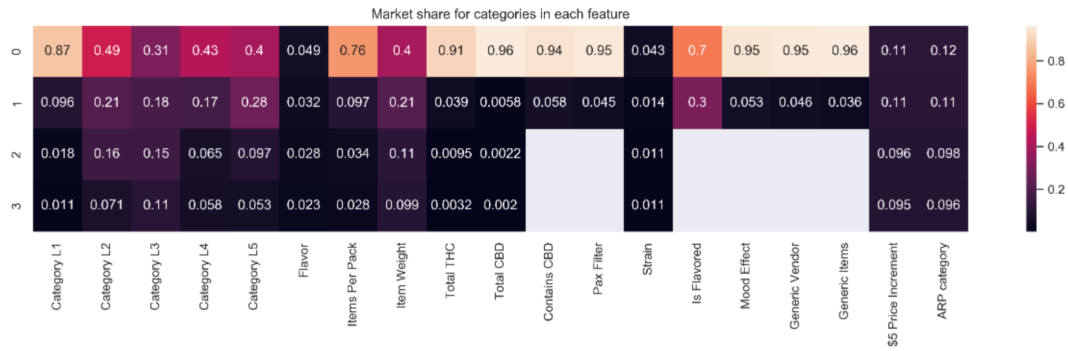


Figure 14. Heatmap of market share for major values of each feature

We also construct a data frame for *out performance* likewise, plotted below:

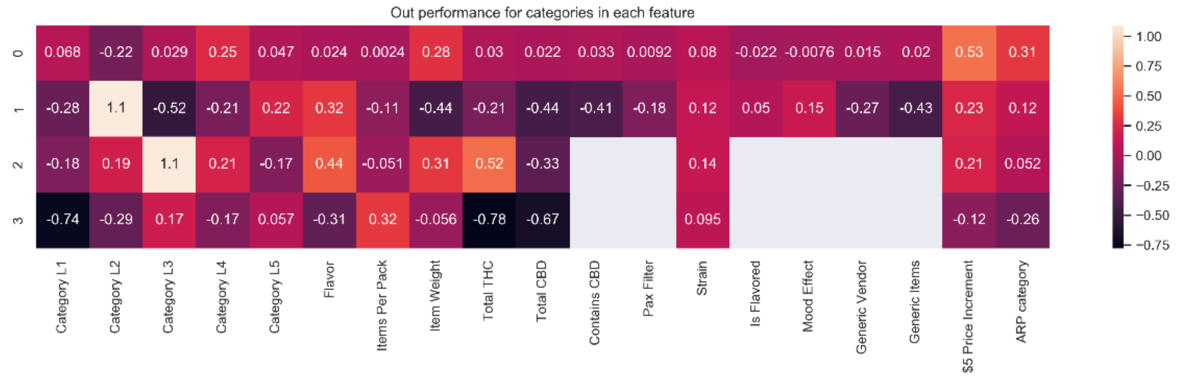


Figure 15. Heatmap of *out performance* coefficients for major values of each feature

According to the figure 15, we can see that the most powerful indicators are “Flower” in Category L2 and “Hybrid” in Category L3.

5 Discussion

5.1 Intuition on the predictive model

5.1.1 Why is predicting sales for small brand more difficult?

Simply judging from the result of our linear model, we found that it is much more difficult to predict brands with relatively low sales than brands with large sales. This is definitely related to the nature of business: small companies and small brands will have greater volatility. They may stand out in the market, or fail in market competition. It is rare for a small brand to keep sales unchanged. In comparison, brands with large sales will have a relatively stable consumer group, so they will have relatively stable sales. The central limit theorem tells us: only when the sample size is large enough, the characteristics of group behavior become obvious. The customer base of small brands is more unstable, and the number of samples is also smaller. This should explain why the sales of brands with small sales are more difficult to predict.

5.1.2 Is our model overfit or underfit?

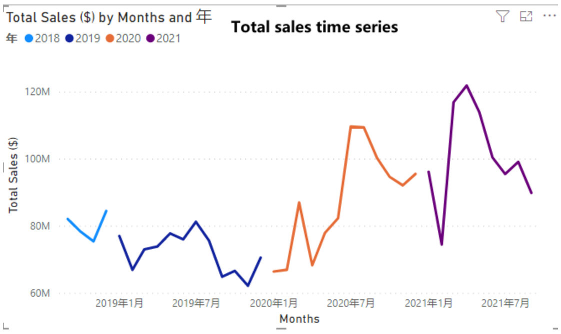
We can see from the results of chapter 3 and chapter 4 that: the prediction of the model learned on a complete data set is better than that of the model that only learns a part of the data set. This shows that the information model learned from other larger brands can also be used to predict the sales of brands with smaller sales. This also illustrates our model may be in a stage of underfitting.

5.2 About how to construct the predictive model

5.2.1 How we deal with the seasonality

We have found that from the figures from Power BI that the sales of the entire market have very obvious seasonality, so we split the time feature into individual year, month, and quarter features, and then perform one hot encoding to capture this seasonality. An alternative method is: Add YoY change as a feature, that is, add the percentage of sales change in the current month compared to the current month of last year. But the problem with this method is that we would use less data. Because our data starts from October 2018. If this method is taken (and drop the rows with null value), we would end up using data starts at October 2019, which will significantly reduce the amount of data available to us. Therefore, we did not choose this method

5.2.1 Why we one-hot encode year feature



We also observed that the sales of the entire cannabis market is rising year over year, so we also choose the year as a feature, and one hot encoding is performed on it. This is equivalent to giving a steady starting level sales for every year. Another method is to ordinal encode the year. In this way, our data can reflect the positive correlation between total sales and year. But the problem with this method is: the increase in sales in the entire market may not have a linear relationship with the year, but may be linearly related to the annual growth rate. Moreover, in linear regression, the input independent variable feature has only four possible values, so it may be difficult to fit a good linear model. Therefore, we did not choose this encoding method. Instead, we chose one hot encoding the year.

5.3 Some ideas on how to further extract information from the dataset

Some ideas about the brands:

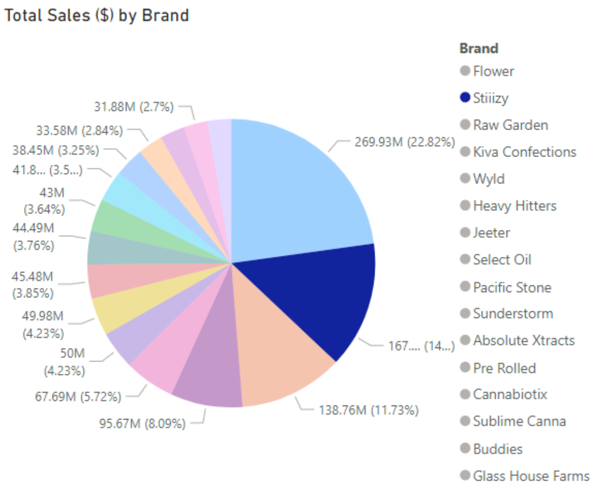
5.3.1 Predict sales by brand or by product?

When we first discussed this question, we found that if we only predict the future sales for the products, then we only have time series data for the TOP50 products. And these time series data may not include some new brands and new products that stands out later in the market. Even if we manage to have good performance on these 50 products, the model may be bad for predicting products with small sales (weak generalization ability, because our the training data set does not contain products with small sales). What's worse, we also wasted the rich timing information contained in *BrandTotalSales.csv*. So, in the end we chose to predict the sales of by brands, not by products.

5.3.2 Why we don't include brand as a feature in predictive model?

The first reason is obvious: we have too many brands! And they should not be ordinal encoded.
The second reason is that <u>brand does not matter that much when market is at its early stage</u>.

We believe that cannabis market is still in the early stages because: 1) sales are still growing rapidly and 2) there are still many brands. Consumers may not have a complete and accurate impression of so many brands, and brand owners have not yet established their own market position as a high-end, mid-range, or low-end brand. When we split the product data by price segment ($5 increment feature in BrandDetail.csv), we found that a brand usually includes products in multiple price ranges.

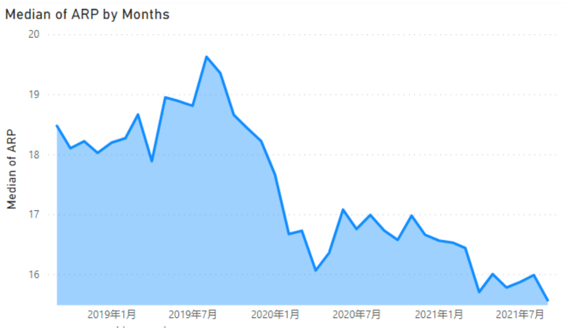


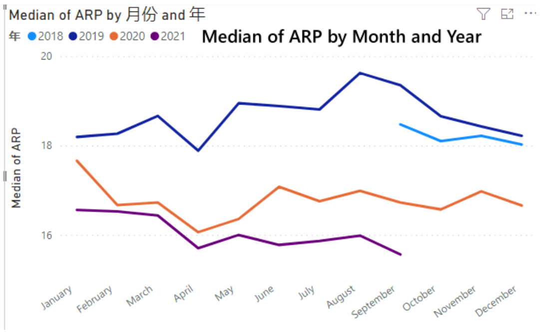
Brand	$5 Price Increment
Stiiizy	$00.00 to $4.99
Stiiizy	$05.00 to $9.99
Stiiizy	$10.00 to $14.99
Stiiizy	$15.00 to $19.99
Stiiizy	$20.00 to $24.99
Stiiizy	$25.00 to $29.99
Stiiizy	$30.00 to $34.99
Stiiizy	$35.00 to $39.99
Stiiizy	$40.00 to $44.99
Stiiizy	$45.00 to $49.99
Stiiizy	$50.00 to $54.99
Stiiizy	$55.00 to $59.99
Stiiizy	$60.00 to $64.99
Stiiizy	$65.00 to $69.99
Stiiizy	$70.00 to $74.99
Stiiizy	$75.00 to $79.99
Stiiizy	$80.00 to $84.99
Stiiizy	$85.00 to $89.99
Stiiizy	$90.00 to $94.99
Stiiizy	$95.00 to $99.99
Stiiizy	Over $100

It shows that this brand does not position itself as a brand in a certain price range. This evidence also supports the assumption that the whole cannabis market is in the early stage and consumers do not pay much attention to the brands. So, in the end, we chose to ignore the brand, a variable that everyone seemed to consider to be very important.

5.3.3 Why we ignore Average Retail Price and Total Units

In the three given tables, there is a relationship that the total sales is equal to the total units multiplied by the average retail price. We analyzed the median of average retail price in the *BrandAverageRetailPrice.csv* and found that it is decreasing over time.





However, the total sales is gradually rising. Therefore, the changes of total sales mainly come from the changes in total units.

In general, the average retail prices in two months remains in a narrow range. Moreover, the average selling price in the market depends on the relationship between supply and demand in the market. We believe that this is not a variable that can be estimated by our model for given information. So, we did not include this variable in the predictive model. We can also find from the steady decline in average retail prices that cannabis market is an emerging market. The supply is constantly increasing, and the rate of increase in demand is not as fast as the rate of increase in supply. Therefore, the average retail price is falling. Another possible explanation is that more and more people are participating in this market, lowing the average retail prices. And this market is becoming a mass market.

5.4 About the key indicator of successful product

If there is a very effective indicator for successful products, then products with positive indicators value should success in the market (monthly sales gradually increase), and vice versa. We originally considered using more time series data to construct the indicators for judging the success of the product. In this way, we can see how the indicator drives sales growth over time. Unfortunately, we only have a few product-level time series data, so we can't do that. Instead, inspired by the bayes theorem, the indicator we use to measure success is the probability of the products represented by this indicator appearing in the excellent sales group. In this way, we can calculate how good a certain category is within a selected feature.

5.5 What can be improved

The category information we use in the predictive model included only the L1 level. This is because deep-level categories have various values that we cannot do one-hot encode on. From the reading materials and some website, we found that the classification use by the dataset is inconsistent with the actual classification of products by store sales. Maybe using the store's classification method for products to encode our training set may achieve better results.

6 Conclusion

In this project, we first extract five time-series features from original dataset, which are total sales of last month, rolling average total sales of last three months, year, month and quarter of sales respectively. Besides, category information under brands is used to create additional features. Then, we implement and execute a comprehensive pipeline to handle with raw data and obtain a prepared dataframe. Before training model, PCA is implemented to decrease dimensionality of dataframe. and GridSearchCV method is employed to select the optimal parameters of models. Next, three different models (Linear Regression, Random Forest and XGBoost model) are implemented to forecast total sales of each brand for September. Moreover, we tired segmented regression, which means that split the prepared dataframe into two parts and perform model training on these two parts separately. The prediction performance metrics of all models are reported and compared, the result shows that the highest performing model is XGBoost modelIn this project, we first extract five time series features from original dataset, which are total sales of last month, rolling average total sales of last three months, year, month and quarter of sales respectively. Besides, category information under brands is used to create additional features. Then, we implement and execute a comprehensive pipeline to handle with raw data and obtain a prepared dataframe. Before training model, PCA is implemented to decrease dimensionality of dataframe. and GridSearchCV method is employed to select the optimal parameters of models. Next, three different models (Linear Regression, Random Forest and XGBoost model) are implemented to forecast total sales of each brand for September. Moreover, we tired segmented regression, which means that split the prepared dataframe into two parts and perform model training on these two parts separately. The prediction performance metrics of all models are reported and compared, the result shows that the highest performing model is XGBoost.

In the second part, we construct several indicators, such as ‘market share’ and ‘out performance’ coefficient to measure the relationships between certain value of selected feature and total sales. And then we ran those measures on every possible value for every useful feature and found that the most powerful two indicators of successful product are “Flower” in Category L2 and “Hybrid” in Category L3.