Forecasting E-Commerce Sales
ETM 58D Term Project

Prepared by
"Group 4"
Sadettin Ayhan Ertuğlu
Boray Yurdakul

for
ETM 58D / Business Analytics
Engineering and Technology Management Department

July 2020

Table of Contents

Table of Figures

1. Introduction

For the project of the ETM 58D Business Analytics course, it is asked to forecast e-commerce sales of certain products on Trendyol. One of the main aspects of business analysts is making forecasting based on analyzing definite data and/or trends. With the increase in the interest in such concepts like data science, and the results of the valuable outcomes of the forecasts, the significance of the forecasting and ability of analyze data, forecasting become a crucial tool for multiple industries. For instance, a factory may forecast its expansion to buy new machines, land and employee or an e-commerce company may forecast its sales to manage its inventory and infrastructure more efficiently. Successful predictions for forecasts result in good management of company. Companies that realize this advantage started to invest in business analysts in recent years. Thanks to them and their analysis of both past and current data, companies can see future less blurry. There are certain methods of forecasting such as, naïve approach, time series methods, drift method[1]. In this report, five of them*, naïve approach, linear regression, gradient boosting, cv penalized regression and penalized regression,* are employed to test which one is best in our case to forecast the e-commerce sales counts.

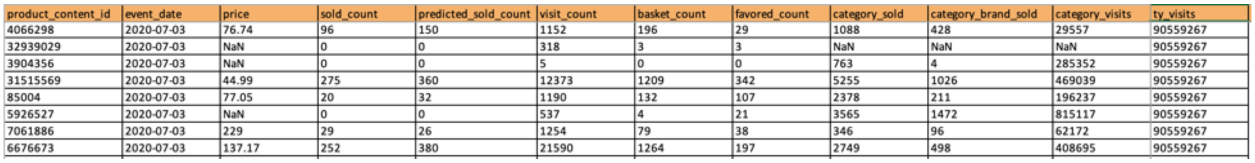
There was also a competition between 6 teams in the course, where our predictions are compared with each other regarding to the real sold counts. This competition started on 15th of June and ended on 5th of July. For this time period data provided by Trendyol is updated on daily basis in each night.

For all forecasting, coding and plotting purposes R Software has been employed.

[1] https://en.wikipedia.org/wiki/Forecasting

2. Descriptive Data Analysis

As a co-operation with Trendyol, *one of the most popular shopping web sites in Turkey with tens of millions of daily views,* 8 items' different parameters are provided for the time period of between 2019-05-13 and 2020-07-06. In the figure below, a representation of the given data can be found. Also, in the following table, product ID descriptions can be found.



product_content_id	event_date	price	sold_count	predicted_sold_count	visit_count	basket_count	favored_count	category_sold	category_brand_sold	category_visits	ty_visits
4066298	2020-07-03	76.74	96	150	1152	196	29	1088	428	29557	90559267
32939029	2020-07-03	NaN	0	0	318	3	3	NaN	NaN	NaN	90559267
3904356	2020-07-03	NaN	0	0	5	0	0	763	4	285352	90559267
31515569	2020-07-03	44.99	275	360	12373	1209	342	5255	1026	469039	90559267
85004	2020-07-03	77.05	20	32	1190	132	107	2378	211	196237	90559267
5926527	2020-07-03	NaN	0	0	537	4	21	3565	1472	815117	90559267
7061886	2020-07-03	229	29	26	1254	79	38	346	96	62172	90559267
6676673	2020-07-03	137.17	252	380	21590	1264	197	2749	498	408695	90559267

Figure 1. Given Data by Trendyol

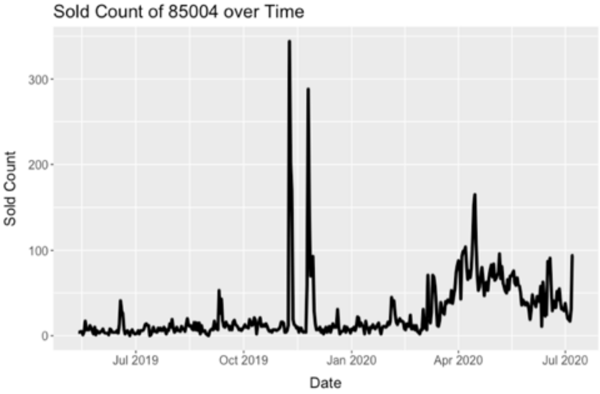
Table 1. Product ID Descriptions

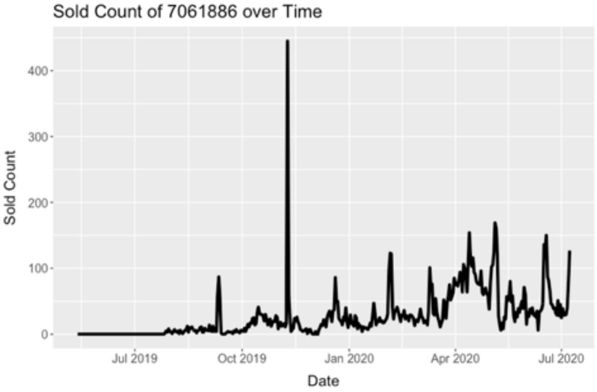
Product ID	Brand	Description
4066298	Sleepy	Islak Havlu Mendil Yenidoğan Naturel 12x40 Paket 480 Yaprak
32939029	Oral-B	Vitality 100 Cross Action White Elektrikli Diş Fırçası
3904356	Koton	Erkek Lacivert Fermuar Detaylı Kapitone Mont
31515569	Trendyolmilla	Siyah Yüksek Bel Toparlayıcı Örme Tayt
85004	La Roche Posay	Effaclar Yüz Temizleme Jeli
5926527	Trendyolmilla	Siyah Kaplı Çiçek Desenli Bikini Üstü
7061886	Fakir	Lucky Dikey Elektrikli Süpürge - Kum Beji
6676673	Xiaomi	Redmi Airdots Tws Bluetooth Basic 5.0 Kulaklık

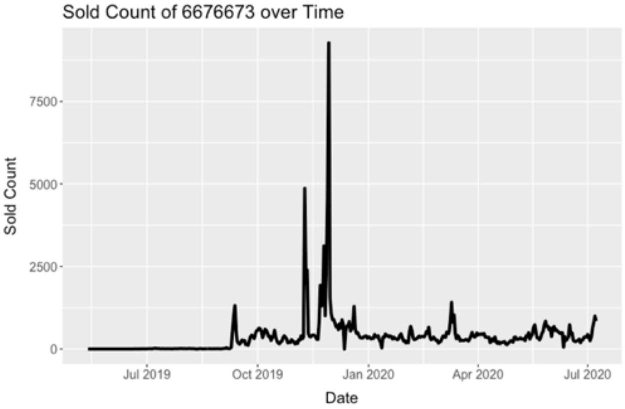
When we look at the data first, we see that data is provided from 2019-05-13 till present day and updated till the last day of the competition. At first, it is realized that prices are very fluctuated during period according to financial changes in the country and based on sales in special days such as new year, black Friday, holidays etc.

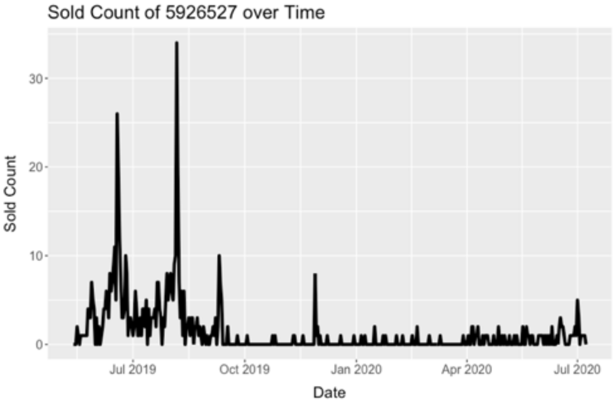
Also, since there are NAs in data, it is tried to manipulate them and get rid of NAs. First, KNN imputation is utilized to fill the NA values, but R gave a warning that there isn't enough data to fill NA values with KNN imputation method. To explain briefly how KNN imputation works, it is a row-based method and finds the closest point (row) based on the k neighbors and take mean of the neighbors, or weighted mean, where the distances to neighbors are used as weights, so the closer neighbor is, the more weight it has when taking the mean. Using weighted mean seems to be used most commonly. After KNN imputation didn't work, we filled the NA values with column mean. It is important to fill NA values reasonable, since we analyze product separately, NA values were filled product by product.

In the figure set below, sold counts of all products over time can be found. As can be observed, the data is quite fluctuating and hard to read.









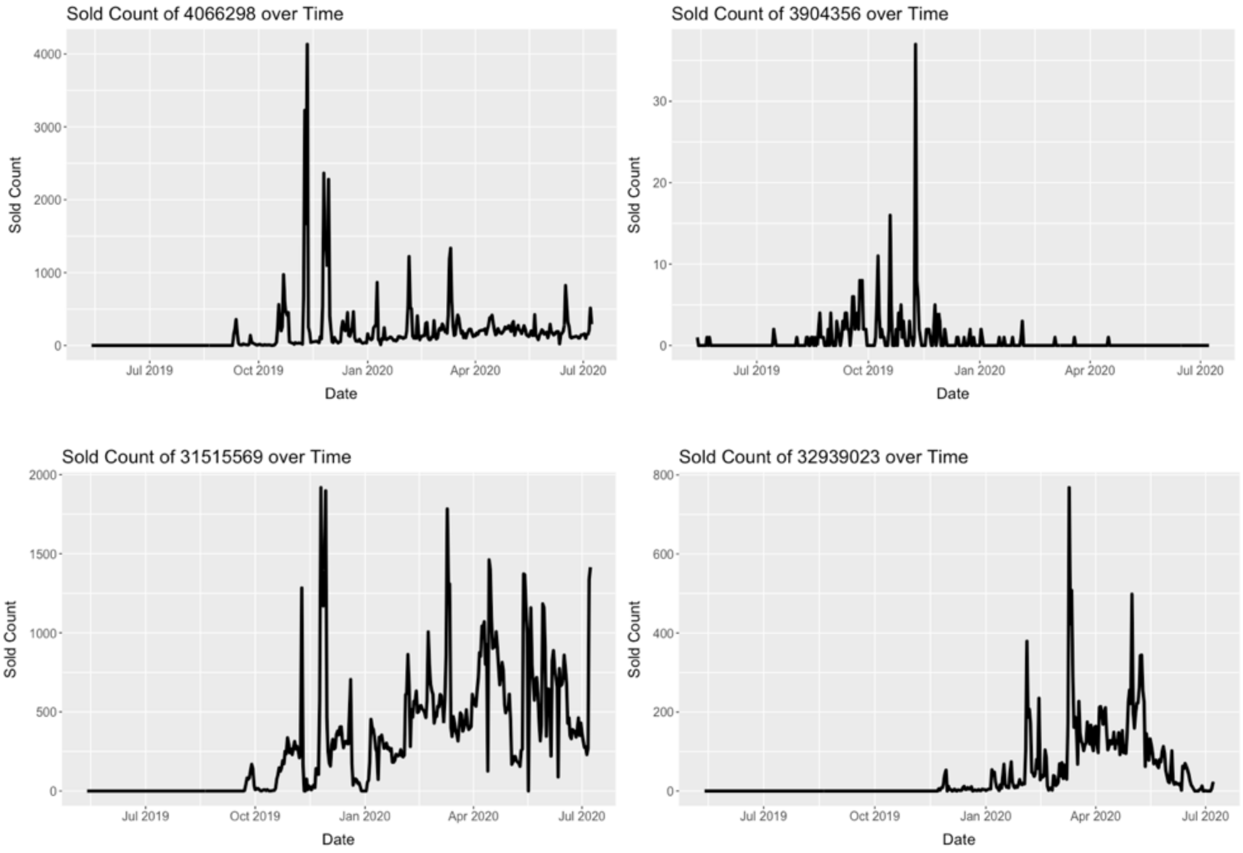
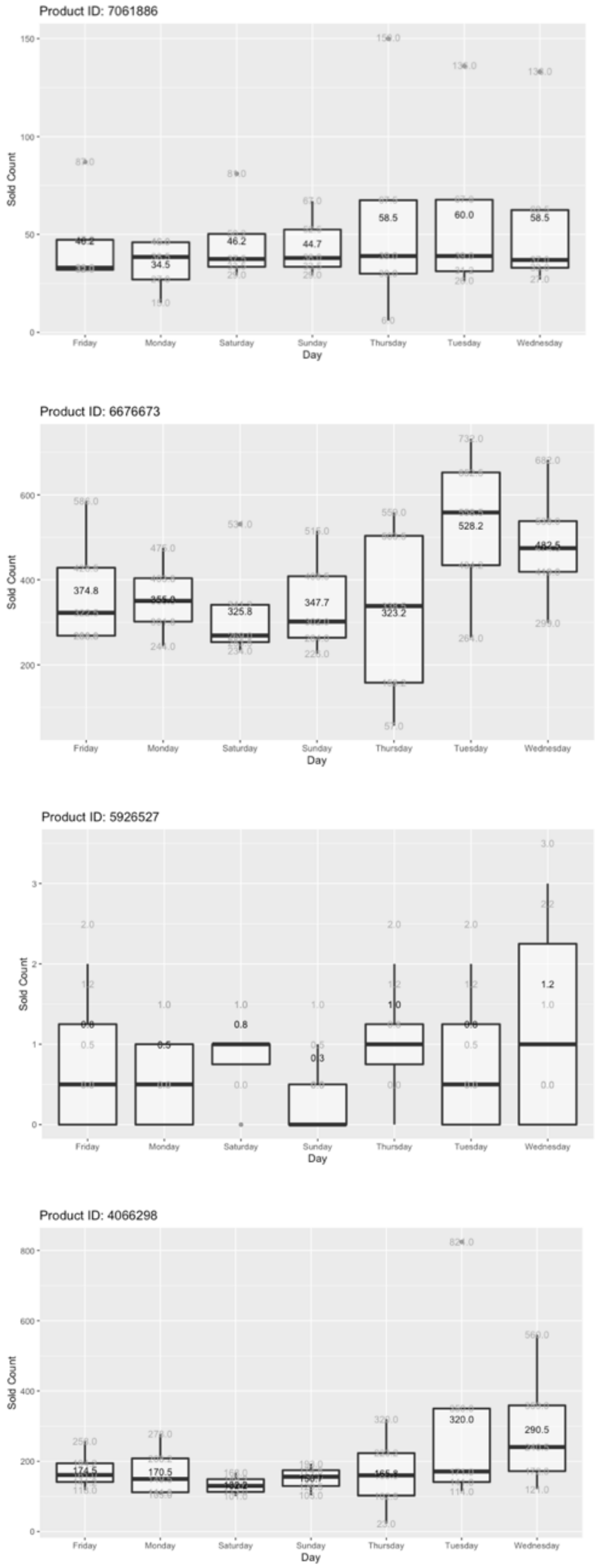


Figure 2. Sold Count vs Time Graphs of each Product

There are two seasonal products in the data which are coat and the bikini top. As can be seen from the plots above, sold count of those products were almost zero for all over the forecast period. Not only the seasonality but also stock count is affected the sold count of those products. The coat was sold out almost every day since January. Moreover, there was limited stock amount (only size 34) for the bikini top during the forecast period.

Moreover, to observe whether there is a relationship between the days of the week and the sold counts, two boxplots were drawn. While Figure 3 is plotted with the last month's sold counts, Figure 4 is plotted with the last week's sold count. At the day where the Figure 4 was plotted, when it is going back 7 days and it collided with two Saturdays, for that specific day, a box shaped is generated. On the other hand, the other days are generated with a single day's data.



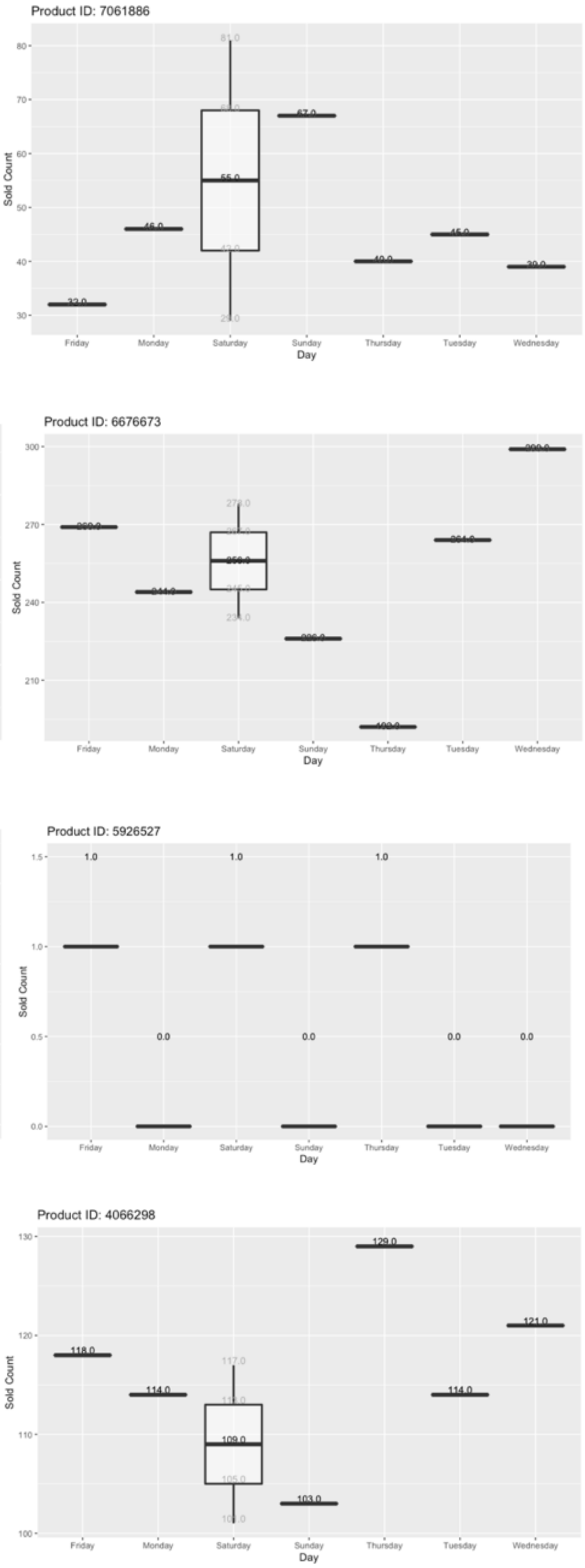


Figure 3. Daily Based Sold Counts for One Month

Figure 4. Daily Based Sold Counts for One Week

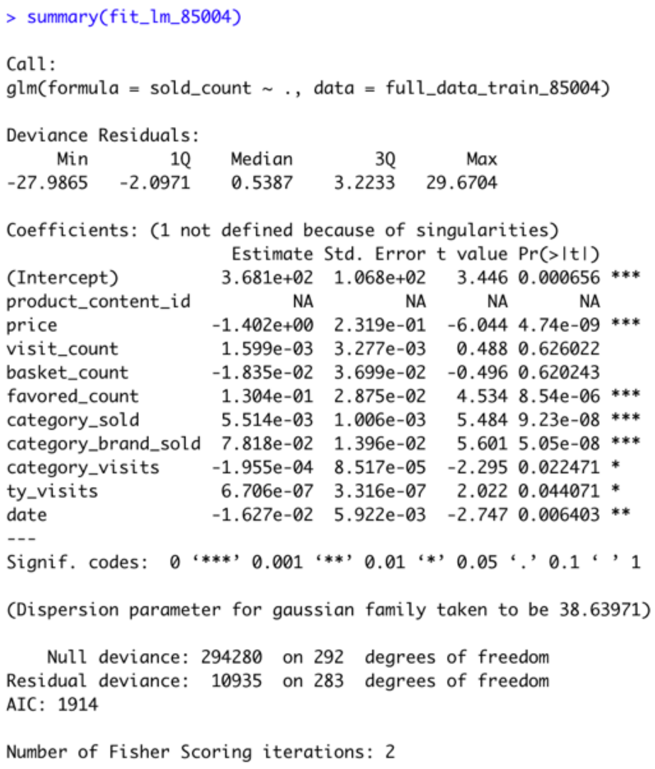
3. Methods

3.1. Naive Approach

At the beginning of the project, we couldn’t have the chance the build a proper model to make predictions. In first days of the competition, we submitted the sold counts one week ago for all products. We build a model for lag_40 because for example, when we were predicting the sales for 30th of June, we have the data till 28th of June. So, we took the sold count values from 5 days ago.

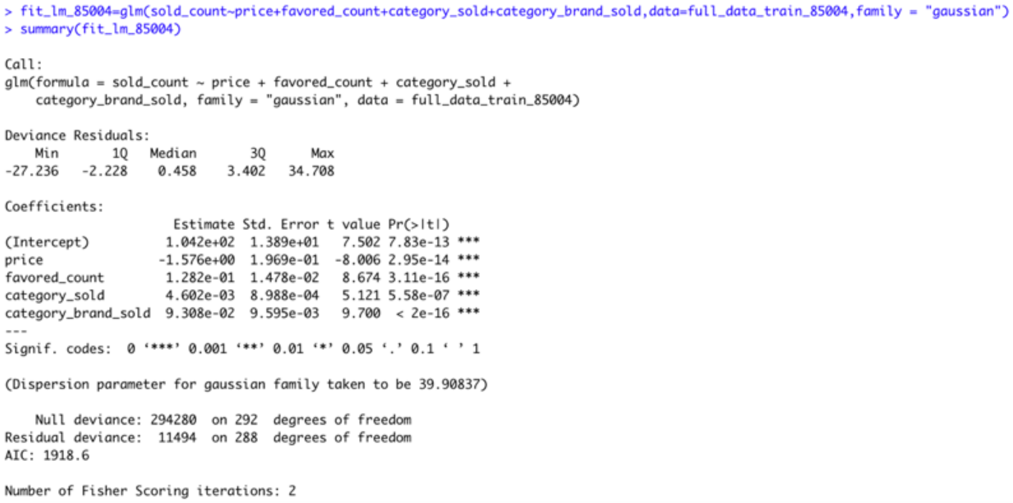
3.2. Linear Regression

In linear regression method generalized linear regression model was used for all products separately. For example, the steps for one product 85004; firstly, model with generalized linear regression was created including all the inputs and we checked the model’s result (Figure 5). Then, as illustrated in Figure.6, the inputs that are the most significant were used to build a new model. When the first model was built with all inputs, MAPE for product 85004 was 29.67 but after we chose the most significant inputs and build a model, MAPE decreased to 24.42. For all products, MAPE decreased by input eliminations. Also, for some of the products, difference between MAPEs were huge. MAPE values for linear regression are illustrated in Figure.7.



```
> summary(fit_lm_85004)

Call:
glm(formula = sold_count ~ ., data = full_data_train_85004)

Deviance Residuals:
     Min        1Q    Median        3Q       Max  
-27.9865   -2.0971    0.5387    3.2233   29.6704  

Coefficients: (1 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          3.681e+02  1.068e+02   3.446 0.000656 ***
product_content_id          NA         NA      NA       NA    
price               -1.402e+00  2.319e-01  -6.044 4.74e-09 ***
visit_count          1.599e-03  3.277e-03   0.488 0.626022    
basket_count        -1.835e-02  3.699e-02  -0.496 0.620243    
favored_count        1.304e-01  2.875e-02   4.534 8.54e-06 ***
category_sold        5.514e-03  1.006e-03   5.484 9.23e-08 ***
category_brand_sold  7.818e-02  1.396e-02   5.601 5.05e-08 ***
category_visits     -1.955e-04  8.517e-05  -2.295 0.022471 *  
ty_visits            6.706e-07  3.316e-07   2.022 0.044071 *  
date                -1.627e-02  5.922e-03  -2.747 0.006403 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 38.63971)

    Null deviance: 294280  on 292  degrees of freedom
Residual deviance:  10935  on 283  degrees of freedom
AIC: 1914

Number of Fisher Scoring iterations: 2
```

Figure 5. Model with All Inputs



```
> fit_lm_85004=glm(sold_count~price+favored_count+category_sold+category_brand_sold,data=full_data_train_85004,family = "gaussian")
> summary(fit_lm_85004)

Call:
glm(formula = sold_count ~ price + favored_count + category_sold + 
    category_brand_sold, family = "gaussian", data = full_data_train_85004)

Deviance Residuals:
    Min       1Q   Median       3Q      Max  
-27.236   -2.228    0.458    3.402   34.708  

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          1.042e+02  1.389e+01   7.502 7.83e-13 ***
price               -1.576e+00  1.969e-01  -8.006 2.95e-14 ***
favored_count        1.282e-01  1.478e-02   8.674 3.11e-16 ***
category_sold        4.602e-03  8.988e-04   5.121 5.58e-07 ***
category_brand_sold  9.308e-02  9.595e-03   9.700  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 39.90837)

    Null deviance: 294280  on 292  degrees of freedom
Residual deviance:  11494  on 288  degrees of freedom
AIC: 1918.6

Number of Fisher Scoring iterations: 2
```

Figure 6. Model with Most Significant Inputs

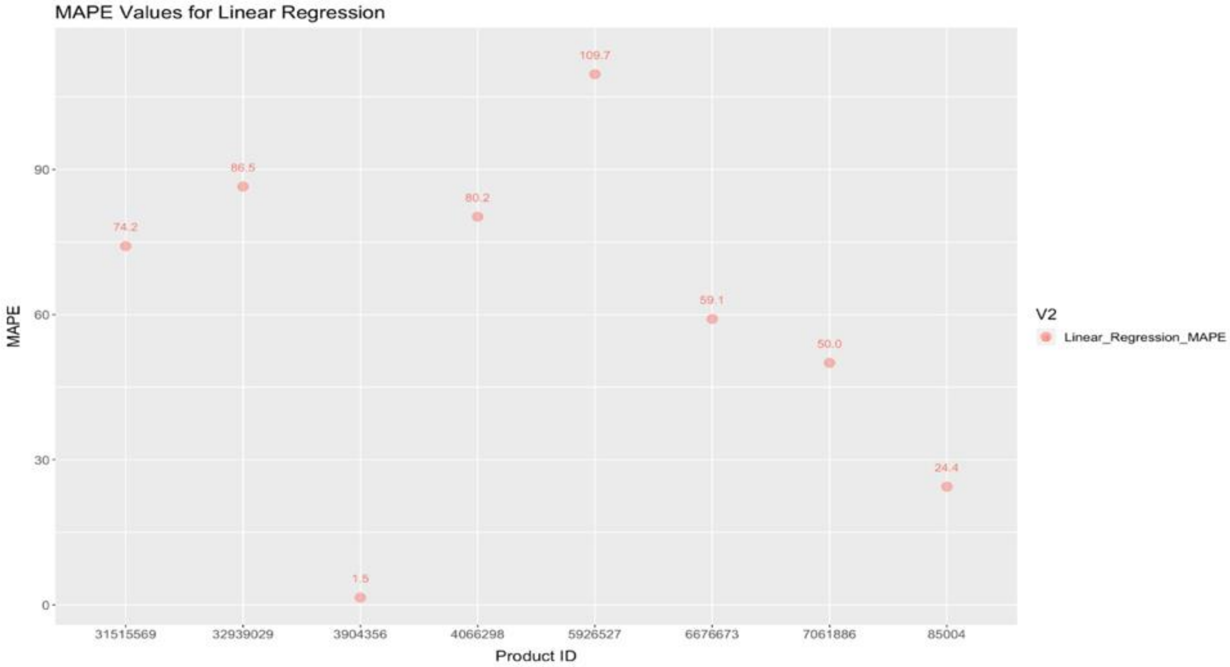
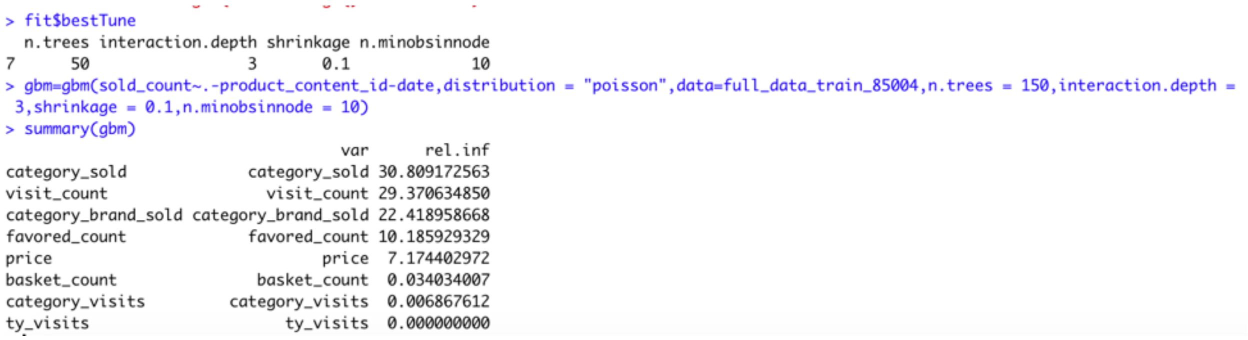


Figure 7. MAPE Values for Linear Regression

3.3. Gradient Boosting

While methods have been searched to be used in the project, it is found out that gradient boosting is a very popular method in Kaggle. It is decided to use gradient boosting to see if it is effective or not. For all products, gradient boosting was used with 10-fold cross validation. K-fold cross validation method is used because it is discussed in class, as it is best and fastest method to minimize error rate. The steps for product 85004 to be an example; As mentioned before, 10-fold gradient boosting was used to find the best features for our model. After the boosting operation, R gave us the best fit as n.trees = 50, interaction.depth = 3, shrinkage = 0.1, n.minobsinnode = 10 (Figure.8). Before using the method for prediction, we analyzed how inputs effect on model in gbm. As illustrated in the upper part, the most significant inputs are category_sold, secondly visit_count and it goes like that. Moreover, basket_count, category_visits and ty_visits have very little effect on predicting sold_count (Figure.8 - Figure.9). For all products, same processes were followed. R gave us the best tunes for each product, and we built the models with the best tunes. MAPE values for gradient boosting are illustrated in Figure.10.



```
> fit$bestTune
  n.trees interaction.depth shrinkage n.minobsinnode
7      50                 3       0.1             10
> gbm=gbm(sold_count~.-product_content_id-date,distribution = "poisson",data=full_data_train_85004,n.trees = 150,interaction.depth =
 3,shrinkage = 0.1,n.minobsinnode = 10)
> summary(gbm)
                                    var      rel.inf
category_sold             category_sold 30.809172563
visit_count                 visit_count 29.370634850
category_brand_sold category_brand_sold 22.418958668
favored_count             favored_count 10.185929329
price                             price  7.174402972
basket_count               basket_count  0.034034007
category_visits         category_visits  0.006867612
ty_visits                     ty_visits  0.000000000
```

Figure 8. Best Tune and Input Analysis

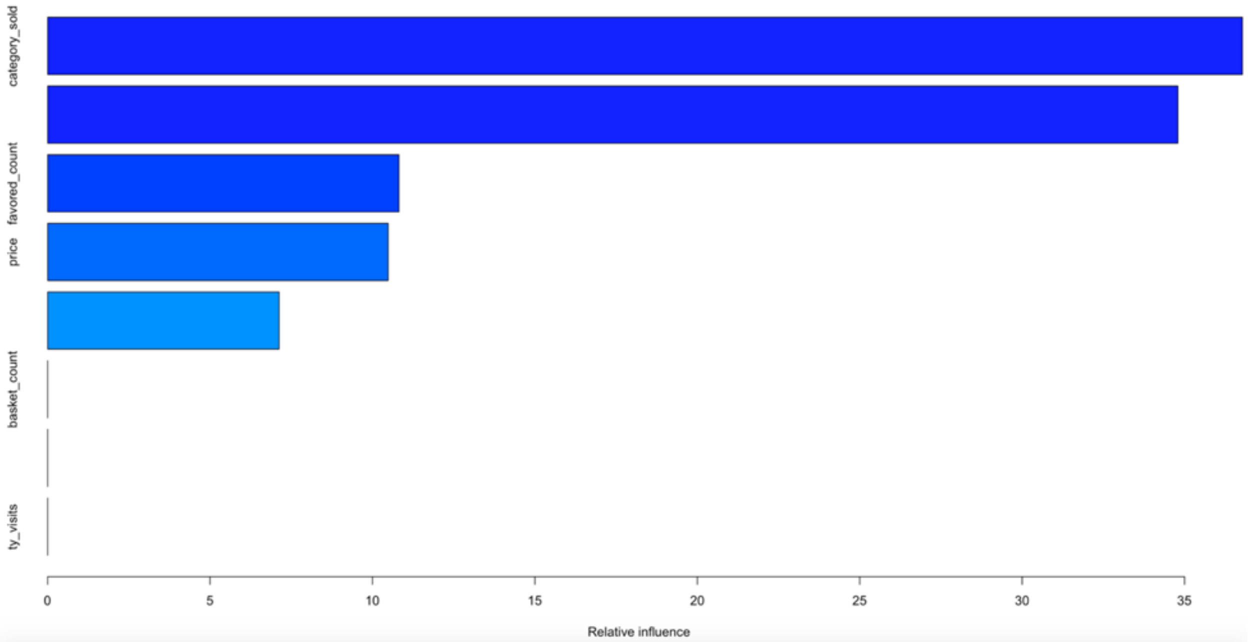


Figure 9. Input Analysis

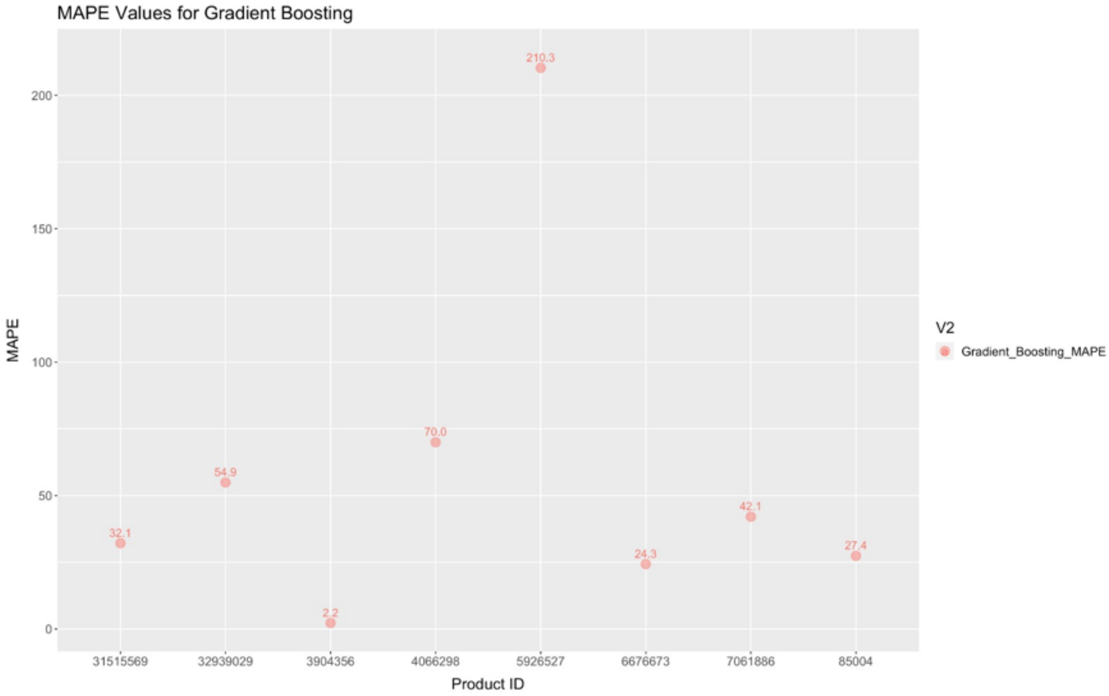
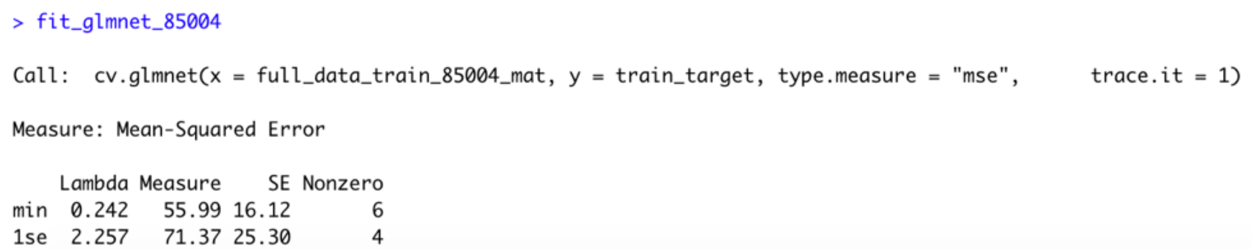


Figure 10. MAPE Values for Gradient Boosting

3.4. CV Penalized Regression

As it is known from the class and homework, there are several methods for penalized regression. In this part, cross validation penalized regression was used. Data was split into training and test sets and converted them to matrixes since glmnet only works with matrixes. For measurement type mean-squared error was determined had the output in Figure 11. Then, it was decided use to lambda.min to build our model for all products since it's mse was smaller than the other lambda (Figure.11). MAPE values for CV penalized regression are illustrated in Figure.12.



```
> fit_glmnet_85004

Call:  cv.glmnet(x = full_data_train_85004_mat, y = train_target, type.measure = "mse",      trace.it = 1)

Measure: Mean-Squared Error

    Lambda Measure    SE Nonzero
min  0.242   55.99 16.12       6
1se  2.257   71.37 25.30       4
```

Figure 11. Summary of CV Penalized Regression

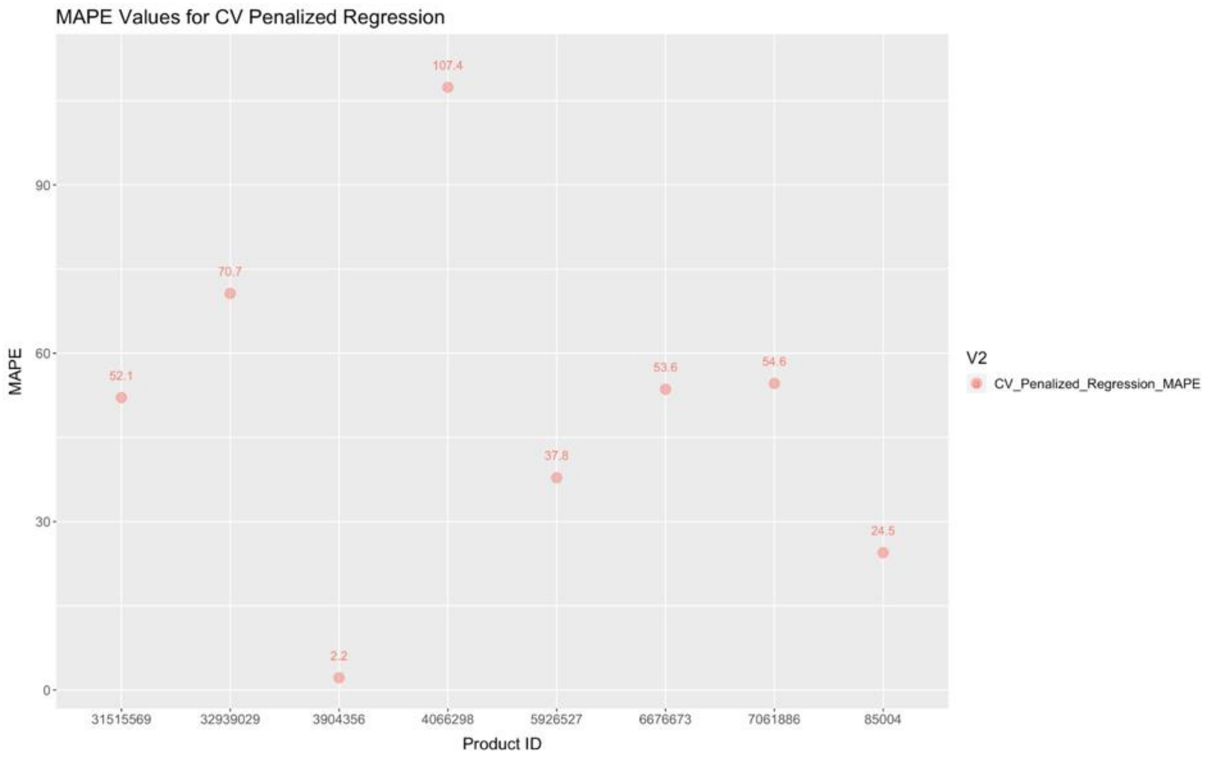


Figure 12. MAPE Values for CV Penalized Regression

3.5. Penalized Regression

Finally, penalized regression gave the best overall MAPE. It is the best way to build a model to make predictions. Glmnet was used to build a model with penalized regression. Glmnet uses default 100 lambda values to build models. In Figure.13 for product 85004, it is illustrated that how many coefficients glmnet used for 100 lambda values and models. After that, SSE was calculated for all 100 lambda values and plot them as in the graph in Figure.14. Lambda 65 was chosen because it has the smallest SSE among all 100 lambda values as shown in Figure.14. Also, it was checked that what lambda 65's value is and what coefficients lambda 65 use to build a model. It can be seen that lambda 65's value and coefficients from the R code. For all products, the same processes were followed and used the best lambda value for them. MAPE values for penalized regression are illustrated in Figure.15.

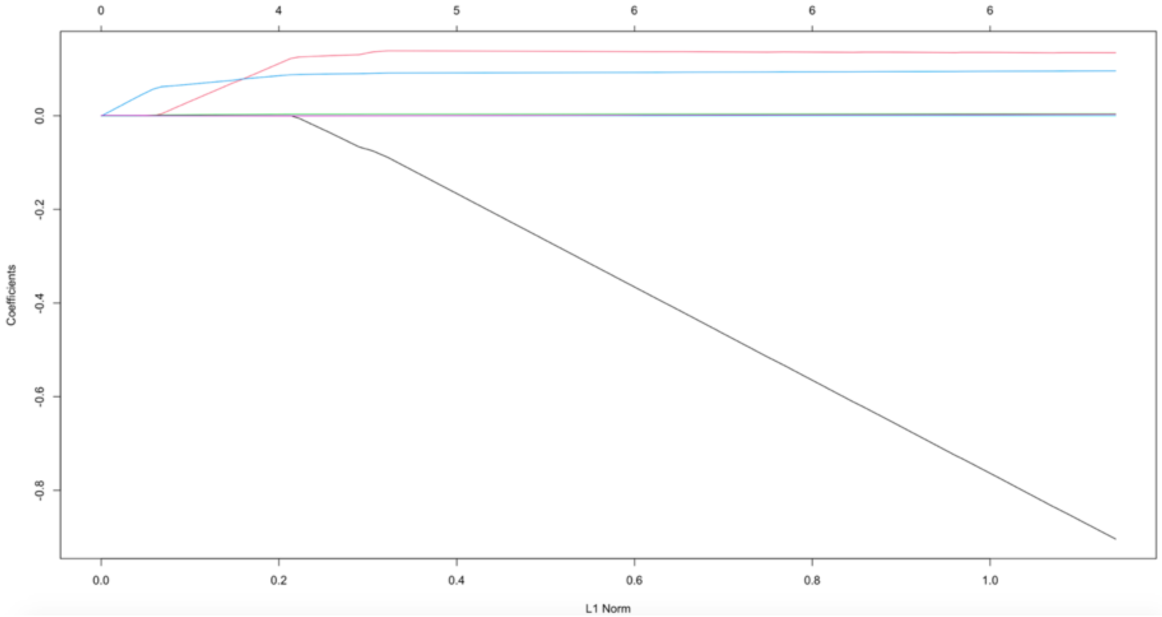


Figure 13. Coefficient Graph

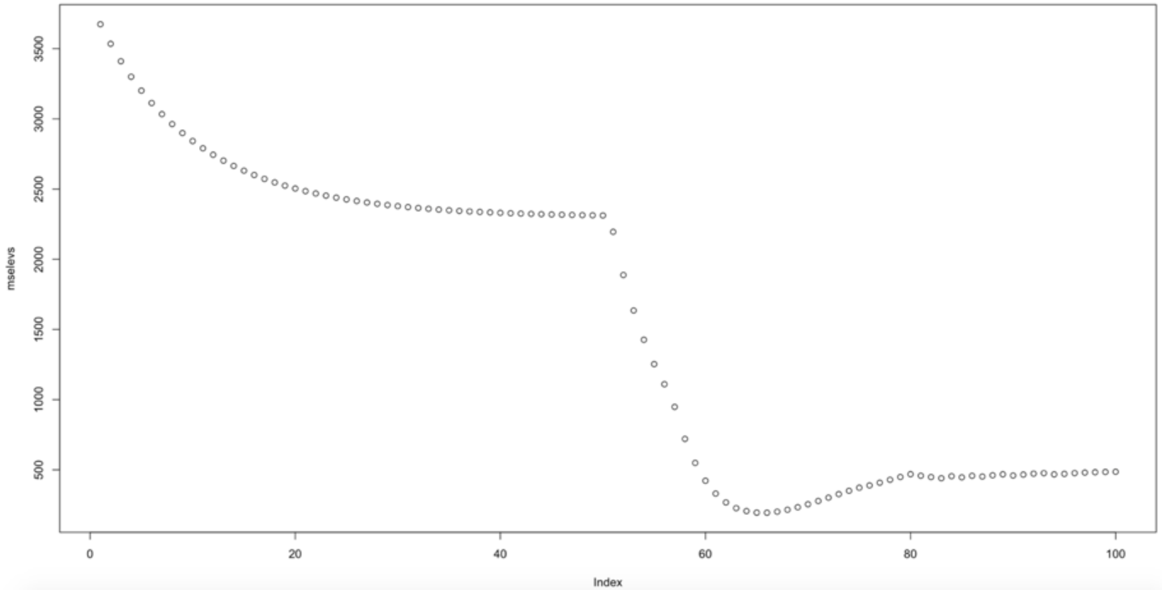


Figure 14. SSE for 100 Lambdas

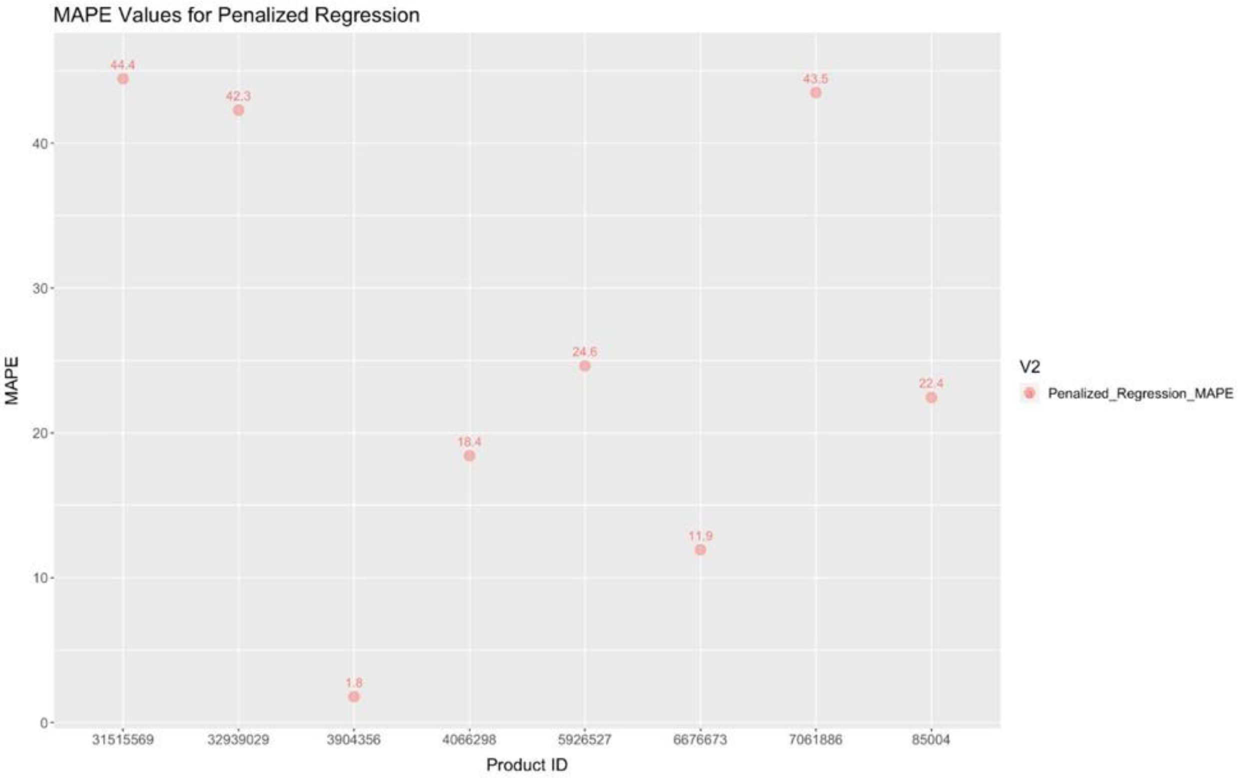


Figure 15. MAPE Values for Penalized Regression

4. MAPE Overview

To compare different methods to see which gives the best results, their MAPE values are compared with each other. To do that, product by product, MAPE values for each method are presented in a single plot as can be seen in Figure.16. As a result, it is observed that linear regression method gave the worst results among all of the methods for almost each product. On the other hand, penalized regression method gave the minimum MAPE values for almost all products except the product 31515569. That’s why, it is decided to use penalized regression method to forecast sold counts of all products.

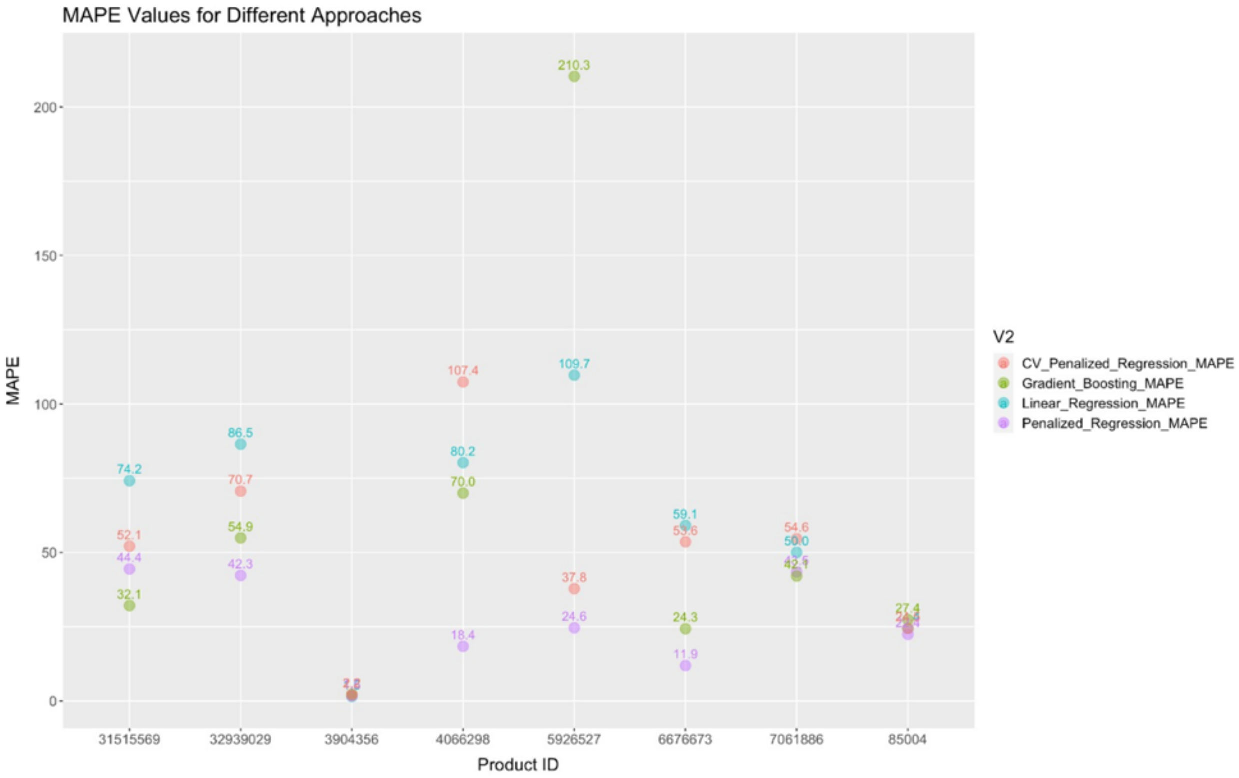


Figure 16. MAPE Overview

5. Submission Method

After the examination of all methods were completed, it was decided that penalized regression has the lowest MAPE value and used penalized regression in all submissions. But before submissions, we did some extra work too. Firstly, after building the model, all predictions were collected to a matrix which is called sonuc. In sonuc matrix, the sold_counts from 1 week ago were taken for analyzing. After that, each product's price was checked before submission on Trendyol. Also, the sold_count value which has the same price with the closest date was checked. Also, the decrease or increase difference was checked between the last day of the given data and the date we are going to predict. For example, when it is asked to predict the sales on 30th of June, the data till 28th of June is published, so we checked the decrease in 21st of June to 23rd of June or the week before that. Finally, after all the information were gathered from the model and past data, we did our submissions. Additionally, social events, such as common salary days, university or high school entry exams, Ramazan Eid days were taken into account while doing the submissions.

6. Limitations and Restrictions

- **Pandemic Period – Excessive Workload and Projects**

Our main limitation was that both of our companies have increased our workload because of the pandemic period.

- **Small data set**

Large data set is important in building models. When we split data into one product, there were around 300 observations. Moreover, data has NA values. Maybe more data would help us to build a better model.

- **Pandemic Period – Change in Customer Shopping Behavior**

In pandemic period, people's shopping behavior changed and focused on mostly essential items for their livings.

- **API Website Shutdown**

Even though it was wanted to place submission vs real sold count figures in the report, the API Website was closed in the recent days. The submissions were not able to be achieved. That's why, predictions and the real sold counts couldn't be compared.

7. Possible Future Work

- **ARIMA & Fourier Transform – Time Series**

Such methods are found on internet which are called ARIMA and Fourier Transform. We think that hey can be used because in e-commerce, time series are very important. Those two methods are being used for time series problems and predictions.

- **Grid Search & Random Search – Parameter Tuning**

Grid search and random search are important aspects to find right inputs to build a good model.

- **Random Forest**

We were working on model evaluation and it gave an output that RF is a good method. We didn't have the chance to develop our model evaluation part properly.

- **PCA**

In our work, we didn't delete our outliers in data. In future, if it is wanted to ignore the outliers and manipulate the data, PCA can be used as we learned in class. PCA helps us to take the data based on variance percentages.

8. Conclusion

All in all, after trying naïve approach, linear regression, gradient boosting, cv penalized regression and penalized regression methods to forecast the 8 items' sold counts on Trendyol, it is decided that the best method is penalized regression. On the other hand, not only the penalized regression is applied before the submissions. Web pages of products are checked before each submit to make sure that the product price is still the same, it has the same amount of sizes or stock as before. Also, daily based sold counts are checked to find a relation between the day changes and the sold counts, for example to observe if the sold count increases or decreases from Wednesday to Friday. In addition to them, social events, such as common salary days, university or high school entry exams, Ramazan Eid days are taken into account while doing the predictions. It was a great experience to reflect the acquired acknowledge from the courses to a real-life scenario. It was fun to use R and learn different approaches to different problems.