Karolis Liubavicius (1671752)

k.liubavicius@student.han.nl

Data Modelling

Table of Contents

[Introduction 2](#_Toc150525684)

[PART 1 2](#_Toc150525685)

[Data Preprocessing 2](#_Toc150525686)

[Regression Models 3](#_Toc150525687)

[Outlier Detection and Handling 4](#_Toc150525688)

[Exploratory Data Analysis (EDA) 7](#_Toc150525689)

[Checking for multicollinearity 8](#_Toc150525690)

[Linear Regression plot 8](#_Toc150525691)

[Comparing the models and selecting the best-fitting one 10](#_Toc150525692)

[PART 2 12](#_Toc150525693)

[Models Predicting Changes in Sales 12](#_Toc150525694)

[Advertising vs. Influencer vs. Customer Satisfaction 14](#_Toc150525695)

[Predicting with a Final Model 14](#_Toc150525696)

[Non-imputed model vs. Imputed model 15](#_Toc150525697)

[Conclusion 16](#_Toc150525698)

[Appendix 18](#_Toc150525699)

## **Introduction**

The primary goal of this analysis is to identify the key factors of sales performance in the given business unit. By applying regression analysis techniques, the assignment aims to show the intricate relationships between financial variables and sales outcomes. The importance of the task is its potential to provide actionable insights for business strategists and decision-makers, helping in optimizing decision-making and, ultimately, enhancing performance.

# **PART 1**

## Data Preprocessing

Describe the initial steps taken to clean and prepare the data for analysis.

Include Python or R code snippets for data preprocessing.

The first and one of the most important parts is Data Preprocessing, which means cleaning and preparing the dataset for analysis. At first, we started looking for missing values and cleaning them, this meant deleting them.

print(df.isna().sum())

df\_clean = df.dropna()

print(df\_clean.isna().sum())

|  |  |
| --- | --- |
| relative\_sales 0  product\_category 0  market\_research 193  advertising 70  customer\_satisfaction 0  influencer 133  competitors 0  shipping\_time 0  dtype: int64 | relative\_sales 0  product\_category 0  market\_research 0  advertising 0  customer\_satisfaction 0  influencer 0  competitors 0  shipping\_time 0  dtype: int64 |

The second step in Data Preprocessing, finding and defining types of values, in this case, we are looking for numerical and categorical values, and former ones we need to transform them into numerical ones by using dummy variables.

df\_clean = pd.get\_dummies(df\_clean, columns=['product\_category', 'influencer']).astype(int)

df\_clean = pd.concat([df\_clean], axis=1)

relative\_sales 56

market\_research 27

advertising 36

customer\_satisfaction 4

competitors 12

shipping\_time 4

product\_category\_household\_appliances 2

product\_category\_personal\_care 2

product\_category\_small\_electronics 2

product\_category\_tools 2

influencer\_no 2

influencer\_yes 2

dtype: int64 (2326, 12)

## Regression Models

Regression models are specifically designed for tasks where the goal is to predict a continuous numerical output variable (dependent variable) based on one or more input variables (independent variables). the assignment involves analyzing and predicting financial data, which often involves predicting numerical outcomes.

That being said let us dig into the very first model and its results, nothing has been done with the data so far, only NaN values replaced, and there are still possible outliers.

X = df\_clean.drop(columns=['relative\_sales'])

y = df\_clean['relative\_sales']

#constant for independent variables for the intercept in the model

X = sm.add\_constant(X)

#fitting the model

model = sm.OLS(y, X).fit()

print(model.summary())

**Model 1**

**OLS Regression Results**

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.515

Model: OLS Adj. R-squared: 0.513

Method: Least Squares F-statistic: 273.4

Date: Tue, 07 Nov 2023 Prob (F-statistic): 0.00

Time: 12:46:09 Log-Likelihood: -7017.8

No. Observations: 2326 AIC: 1.406e+04

Df Residuals: 2316 BIC: 1.411e+04

Df Model: 9

Covariance Type: nonrobust

===================================================================================

coef std err t P>|t| [0.025 0.975]

-----------------------------------------------------------------------------------

const -1.8968 0.544 -3.487 0.000 -2.964 -0.830

market\_research -0.5947 0.026 -23.160 0.000 -0.645 -0.544

advertising 0.5972 0.021 28.744 0.000 0.556 0.638

customer\_satisfaction 2.2182 0.198 11.216 0.000 1.830 2.606

competitors 0.0007 0.001 0.700 0.484 -0.001 0.002

shipping\_time 0.1239 0.121 1.021 0.307 -0.114 0.362

product\_category\_household\_appliances -0.1133 0.213 -0.533 0.594 -0.530 0.304

product\_category\_personal\_care 0.3811 0.279 1.367 0.172 -0.166 0.928

product\_category\_small\_electronics -1.0481 0.280 -3.748 0.000 -1.596 -0.500

product\_category\_tools -1.1166 0.214 -5.229 0.000 -1.535 -0.698

influencer\_no -4.8004 0.295 -16.280 0.000 -5.379 -4.222

influencer\_yes 2.9035 0.305 9.511 0.000 2.305 3.502

==============================================================================

Omnibus: 661.843 Durbin-Watson: 2.055

Prob(Omnibus): 0.000 Jarque-Bera (JB): 2473.602

Skew: -1.364 Prob(JB): 0.00

Kurtosis: 7.253 Cond. No. 5.75e+17

==============================================================================

The R-squared value is 0.515, which means that approximately 51.5% of the variance in relative sales can be explained by the independent variables in the model. The F-statistic is 273.4, This indicates that at least one of the independent variables has a significant effect on relative sales. In other words, the model as a whole is statistically significant. "advertising" has a positive coefficient of 0.5972, suggesting that an increase in advertising spending is associated with an increase in relative sales. On the other hand, "market\_research" has a negative coefficient of -0.5947, indicating that higher spending on market research is associated with lower relative sales.

## Outlier Detection and Handling

#function to check for impossible values

def check\_impossible\_values(data\_point):

return (

data\_point['competitors'] >= 0 and

data\_point['market\_research'] <= 100 and

1 <= data\_point['customer\_satisfaction'] <= 5 and

data\_point['shipping\_time'] > 0

)

Since most of the data points in the data set are valuable and it is difficult to state that they are outliers and remove them, the first and one of the most important steps was to remove outliers based on logic. For example, in the “competitors” column, the company cannot have a negative number of competitors, or customer satisfaction columns cannot be lower than 1 or higher than 5, while the rating system is from 1-5.

The second approach in handling the outliers is Cook’s, which is a method to identify outliers or influential data points, which can be further investigated to determine if they should be retained or removed from the analysis.

cooks\_d = model.get\_influence().cooks\_distance

n = len(df\_clean)

df\_clean['Outlier'] = cooks\_d[0] > 4/n

Cook’s D retrieves influence statistics from the model. Then create a separate column in the data frame and mark data points that are greater than 4/n which is a threshold. This code identifies potential outliers by flagging observations where Cook's D is relatively high compared to the threshold. In Model 2 you can see the effect of removing illogical outliers from the data set (see Model 2). In Model 3 you can see the result after removing Cook’s D outliers and illogical values (see Model 3).

**Model 2**

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.663

Model: OLS Adj. R-squared: 0.661

Method: Least Squares F-statistic: 449.8

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 11:54:38 Log-Likelihood: -6512.5

No. Observations: 2298 AIC: 1.305e+04

Df Residuals: 2287 BIC: 1.311e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const 0.1801 0.488 0.369 0.712 -0.776 1.136

market\_research -0.6068 0.022 -28.190 0.000 -0.649 -0.565

advertising 0.5681 0.017 32.604 0.000 0.534 0.602

customer\_satisfaction 2.0464 0.166 12.335 0.000 1.721 2.372

competitors -0.2768 0.056 -4.921 0.000 -0.387 -0.166

shipping\_time 0.0298 0.102 0.293 0.770 -0.170 0.230

product\_category\_household\_appliances 0.2704 0.185 1.463 0.144 -0.092 0.633

product\_category\_personal\_care 0.8804 0.236 3.724 0.000 0.417 1.344

product\_category\_small\_electronics -0.2773 0.238 -1.164 0.245 -0.745 0.190

product\_category\_tools -0.6935 0.185 -3.749 0.000 -1.056 -0.331

influencer\_no -3.8137 0.262 -14.556 0.000 -4.328 -3.300

influencer\_yes 3.9938 0.270 14.788 0.000 3.464 4.523

Outlier -13.5108 0.432 -31.299 0.000 -14.357 -12.664

==============================================================================

Omnibus: 165.212 Durbin-Watson: 2.074

Prob(Omnibus): 0.000 Jarque-Bera (JB): 796.955

Skew: 0.113 Prob(JB): 8.78e-174

Kurtosis: 5.876 Cond. No. 6.99e+16

==============================================================================

As you can see, just by removing illogical values the model’s R-squared increased to 0.663 which is 66.3% of the variance in relative sales that can be covered by the independent variables. While Model 3 with removed Cook’s D outliers covers 0.611 or 61.1% of the variance in relative sales. That being said, we can state that Model 2 fits better without removing Cook’s D outliers is more representative, and removing outliers affects the model negatively.

**Model 3**

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.611

Model: OLS Adj. R-squared: 0.609

Method: Least Squares F-statistic: 382.1

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 11:54:38 Log-Likelihood: -6091.0

No. Observations: 2202 AIC: 1.220e+04

Df Residuals: 2192 BIC: 1.226e+04

Df Model: 9

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -0.0209 0.470 -0.044 0.965 -0.942 0.900

market\_research -0.5867 0.023 -25.773 0.000 -0.631 -0.542

advertising 0.5675 0.017 33.889 0.000 0.535 0.600

customer\_satisfaction 2.0083 0.159 12.649 0.000 1.697 2.320

competitors -0.2712 0.054 -5.052 0.000 -0.376 -0.166

shipping\_time 0.0597 0.097 0.613 0.540 -0.131 0.251

product\_category\_household\_appliances 0.3335 0.177 1.887 0.059 -0.013 0.680

product\_category\_personal\_care 0.6546 0.227 2.888 0.004 0.210 1.099

product\_category\_small\_electronics -0.4066 0.230 -1.768 0.077 -0.857 0.044

product\_category\_tools -0.6024 0.177 -3.394 0.001 -0.950 -0.254

influencer\_no -3.8297 0.252 -15.200 0.000 -4.324 -3.336

influencer\_yes 3.8088 0.260 14.653 0.000 3.299 4.319

Outlier 0 0 nan nan 0 0

==============================================================================

Omnibus: 35.117 Durbin-Watson: 2.054

Prob(Omnibus): 0.000 Jarque-Bera (JB): 36.542

Skew: -0.316 Prob(JB): 1.16e-08

Kurtosis: 3.003 Cond. No. inf

==============================================================================

## Exploratory Data Analysis (EDA)

cont\_vars = [

'market\_research',

'advertising',

'customer\_satisfaction',

'relative\_sales'

]

corr\_matrix = df\_clean\_valid[cont\_vars].corr()

**A red and blue squares

Description automatically generated**

market\_research advertising customer\_satisfaction \

market\_research 1.000000 -0.026528 0.035586

advertising -0.026528 1.000000 0.005363

customer\_satisfaction 0.035586 0.005363 1.000000

relative\_sales -0.356317 0.419391 0.144263

relative\_sales

market\_research -0.356317

advertising 0.419391

customer\_satisfaction 0.144263

relative\_sales 1.000000

The strongest correlation exists between advertising and relative sales, with a value of 0.419. This positive correlation indicates that as advertising expenditures increase, relative sales also tend to rise. Conversely, market research exhibits a negative correlation with relative sales (-0.356), suggesting that as market research efforts increase, relative sales tend to decline. Customer satisfaction shows a moderately positive correlation with relative sales (0.144), implying that improved customer satisfaction contributes to some extent to increased sales. These correlations provide valuable insights into the factors influencing sales performance and suggest areas for strategic focus in marketing efforts. Therefore, the values below 0,5 should be considered highly correlated, this means, that the data set is good to be used for modeling.

## Checking for multicollinearity

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

X = df\_clean\_valid[['market\_research', 'advertising', 'customer\_satisfaction', 'relative\_sales']]

X = sm.add\_constant(X)

vif\_data = pd.DataFrame()

vif\_data['Variable'] = X.columns

vif\_data['VIF'] = [variance\_inflation\_factor(X.values, i) for i in range(X.shape[1])]

Variable VIF

0 const 79.620603

1 market\_research 1.182852

2 advertising 1.246682

3 customer\_satisfaction 1.036385

4 relative\_sales 1.468399

The variance inflation factor (VIF) table shows that all the variables in the model have VIF values below 5, which is a safe threshold. This indicates no significant multicollinearity in the model, and the regression results are reliable.

The highest VIF value in the model is for the constant variable (79.620603). This is not a concern, as the constant term is not a predictor variable and does not affect the interpretation of the other coefficients in the model.

The next highest VIF value is for the relative sales variable (1.468399). This slightly elevated VIF value suggests that the relative sales variable is moderately correlated with the other variables in the model. However, this level of correlation is not high enough to cause serious problems with the regression results.

Overall, the VIF table suggests that the regression model is well-specified and has no significant multicollinearity. The regression results can therefore be interpreted with confidence.

Specifically, the VIF value for the relative sales variable is 1.468399. This means that the variance of the relative sales variable is inflated by 46.8399% due to its correlation with the other variables in the model. However, this level of inflation is not high enough to cause serious problems with the regression results.

## Linear Regression plot

**Market Research vs. Relative Sales**

The scatter plot shows a positive linear relationship between market research and relative sales. This means that as market research spending increases, relative sales also tend to increase. The trend line shows that for every additional unit of market research spending, relative sales are expected to increase by about 5 units. However, there is some scatter in the data, which means that there is not a perfect linear relationship. This suggests that other factors, such as the quality of the market research or the overall market conditions, may also play a role in determining relative sales.

A graph of a graph

Description automatically generated with medium confidence

The regression seems quadratic, which means that we have to apply polynomial modeling to market research variables, in this case, we create a new independent variable market\_research\_sqrt.

df\_clean\_valid['market\_research\_sqrt'] = df\_clean\_valid['market\_research'] \*\* 2

**Advertising vs. Relative Sales**

The relationship between advertising and relative sales (see picture 2). The data points are scattered around a line, which indicates that there is a positive linear relationship between the two variables. This means that as advertising spending increases, relative sales also tend to increase. However, the scatter in the data suggests that there is not a perfect linear relationship. This may be due to other factors that influence relative sales. Overall, the plot shows that there is a positive relationship between advertising and relative sales.

A graph of sales

Description automatically generated with medium confidence

**Customer Satisfaction vs. Relative Sales**

The plot shows a positive linear relationship between customer satisfaction and relative sales. This means that relative sales also tend to increase as customer satisfaction increases. The trend line shows that for every additional unit of customer satisfaction, relative sales are expected to increase. However, there is some scatter in the data, which means that there is not a perfect linear relationship. This suggests that other factors, such as the price of the product or the availability of substitutes.

A graph of a customer satisfaction

Description automatically generated

## Comparing the models and selecting the best-fitting one

#Exploring the best-fitting models

#checking last time for the null\_values

print(df\_clean\_valid.isnull().sum())

#model from 1b linear model

print(model\_valid.summary())

#polynominal model containing shipping\_time variable

X\_polynominal = df\_clean\_valid.drop(columns=['relative\_sales']).astype(float)

y\_polynominal = df\_clean\_valid['relative\_sales']

X\_polynominal = sm.add\_constant(X\_polynominal)

model\_polynominal = sm.OLS(y\_polynominal, X\_polynominal).fit()

print(model\_polynominal.summary())

#same polynomial with dropped shipping\_time and since it is not significant (final)

X\_polynominal\_2 = df\_clean\_valid.drop(columns=['relative\_sales', 'shipping\_time']).astype(float)

y\_polynominal\_2 = df\_clean\_valid['relative\_sales']

X\_polynominal\_2 = sm.add\_constant(X\_polynominal\_2)

model\_polynominal\_2 = sm.OLS(y\_polynominal\_2, X\_polynominal\_2).fit()

print(model\_polynominal\_2.summary())

The three OLS regression models presented different sets of independent variables and polynomial features. The first model, with 10 predictors, provides an R-squared value of 0.663, indicating that it explains about 66.3% of the variance in the dependent variable. However, the second and third models, with an additional polynomial feature (market\_research\_sqrt), demonstrate improved R-squared values of 0.817, indicating better explanatory power. The choice between the two models comes down to their additional feature—the square root of market research. Considering simplicity and interpretability, the third model, which includes 10 predictors and the square root of market research, seems preferable. It maintains a high R-squared value, captures non-linear relationships, and avoids excessive complexity. Therefore, the third model suits as a balanced choice, providing a good fit without unnecessary complexity.

# **PART 2**

## Models Predicting Changes in Sales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  | Dependent variable: relative\_sales | | | | |
|  |  |  |  |  |  |
|  | Primary model | Model without illigocal values | Model without Cooks D outliers | Model with polynomial relationships | Final Model |
|  | | | | | |
| const | -1.90\*\*\* | 0.18 | -0.02 | -18.33\*\*\* | -18.33\*\*\* |
|  | (0.54) | (0.49) | (0.47) | (0.55) | (0.55) |
| Market Research | -0.59\*\*\* | -0.61\*\*\* | -0.59\*\*\* | 4.03\*\*\* | 4.03\*\*\* |
|  | (0.03) | (0.02) | (0.02) | (0.11) | (0.11) |
| market\_research\_sqrt |  |  |  | -0.16\*\*\* | -0.16\*\*\* |
|  |  |  |  | (0.00) | (0.00) |
| Shipping time | 0.12 | 0.03 | 0.06 | 0.00 |  |
|  | (0.12) | (0.10) | (0.10) | (0.08) |  |
| Advertising | 0.60\*\*\* | 0.57\*\*\* | 0.57\*\*\* | 0.58\*\*\* | 0.58\*\*\* |
|  | (0.02) | (0.02) | (0.02) | (0.01) | (0.01) |
| Customer Satisfaction | 2.22\*\*\* | 2.05\*\*\* | 2.01\*\*\* | 2.10\*\*\* | 2.10\*\*\* |
|  | (0.20) | (0.17) | (0.16) | (0.12) | (0.12) |
| Competitors | 0.00 | -0.28\*\*\* | -0.27\*\*\* | -0.28\*\*\* | -0.28\*\*\* |
|  | (0.00) | (0.06) | (0.05) | (0.04) | (0.04) |
| Houshold products | -0.11 | 0.27 | 0.33\* | -4.24\*\*\* | -4.24\*\*\* |
|  | (0.21) | (0.18) | (0.18) | (0.17) | (0.17) |
| Personal Care products | 0.38 | 0.88\*\*\* | 0.65\*\*\* | -4.00\*\*\* | -4.00\*\*\* |
|  | (0.28) | (0.24) | (0.23) | (0.21) | (0.21) |
| Small Electronics | -1.05\*\*\* | -0.28 | -0.41\* | -4.95\*\*\* | -4.95\*\*\* |
|  | (0.28) | (0.24) | (0.23) | (0.21) | (0.20) |
| Tools | -1.12\*\*\* | -0.69\*\*\* | -0.60\*\*\* | -5.15\*\*\* | -5.15\*\*\* |
|  | (0.21) | (0.18) | (0.18) | (0.17) | (0.17) |
| Influencer No | -4.80\*\*\* | -3.81\*\*\* | -3.83\*\*\* | -12.94\*\*\* | -12.94\*\*\* |
|  | (0.29) | (0.26) | (0.25) | (0.28) | (0.28) |
| Influencer Yes | 2.90\*\*\* | 3.99\*\*\* | 3.81\*\*\* | -5.39\*\*\* | -5.39\*\*\* |
|  | (0.31) | (0.27) | (0.26) | (0.29) | (0.29) |
|  | | | | | |
| Observations | 2326 | 2298 | 2202 | 2298 | 2298 |
| R2 | 0.52 | 0.66 | 0.61 | 0.82 | 0.82 |
| Adjusted R2 | 0.51 | 0.66 | 0.61 | 0.82 | 0.82 |
| Residual Std. Error | 4.95 | 4.13 | 3.86 | 3.04 | 3.04 |
| F Statistic | 273.39\*\*\* | 449.84\*\*\* | 382.14\*\*\* | 929.50\*\*\* | 1022.90\*\*\* |
|  | | | | | |
|  | \*p<0.1; \*\*p<0.05; \*\*\*p<0.01 | | | | |
| All continuous variables have not been standardized | | | | | |

**The intercept or constant** term in the model represents the expected relative sales when all independent variables are zero, providing a baseline reference point for analysis. In this case, the intercept is -18.33%, indicating the expected relative sales under these conditions.

The coefficient for **Market Research** suggests that a one-unit increase in market research represents to a 4.03% increase in relative sales. This highlights the positive association between investing in market research and the company's sales performance.

The inclusion of the **square root of Market Research** introduces a non-linear relationship, indicating decreasing returns. Specifically, a one-unit increase in the square root of market research corresponds to a 0.16% decrease in relative sales, emphasizing the importance of finding the right balance in the allocation of resources to market research.

**Shipping Time** variable, with a non-significant coefficient (p > 0.05), does not appear to have a statistically significant impact on relative sales.

**Advertising** indicates a positive impact, as a one-unit increase corresponds to a 0.58% increase in relative sales. This emphasizes the importance of effective advertising strategies in positively influencing the company's sales.

**Customer Satisfaction** is a strong positive factor, where a one-unit increase corresponds to a 2.10% increase in relative sales. This highlights the critical role of customer satisfaction in driving sales.

**The Competitors variable** indicates that a one-unit increase in competitors corresponds to a 0.28% decrease in relative sales. It has the negative impact of strong competition on sales and the need for response to competitive forces.

The coefficients for **product categories** indicate their impacts on sales These coefficients provide insights into the specific effects of each product category on relative sales. The better-performing product categories are Personal Care and Small Electronics, which indicate 0.21%

**Influencers**, being influenced by an influencer (Yes) corresponds to a 5.39 unit decrease in relative sales compared to being influenced by no influencer (No). This suggests that relying on influencers may not be as effective as other strategies in driving sales.

In conclusion, the final model highlights the significance of market research, advertising, customer satisfaction, and competition management in influencing relative sales. Careful consideration of product categories and the impact of influencers is essential for making informed business decisions in optimizing sales performance.

## Advertising vs. Influencer vs. Customer Satisfaction

Based on the final regression model, we can assess the importance of advertising, influencer endorsements, and customer satisfaction in driving relative sales. The coefficient for *advertising is 0.58*, indicating that a one-unit increase in advertising corresponds to a *0.58 unit increase* in relative sales. For *influencer endorsements, the coefficient is -5.39*, suggesting that being influenced by an *influencer (Yes)* corresponds to a 5*.39 unit decrease* in relative sales compared to being influenced by no influencer. Finally, *customer satisfaction has a positive coefficient of 2.10*, indicating that a one-unit increase in *customer satisfaction corresponds to a 2.10 unit increase in relative sales*. The coefficient for advertising suggests a more substantial positive impact on sales than the negative impact associated with influencer endorsements. Customer satisfaction also has a positive influence, but the coefficients alone do not allow for a comparison between advertising and customer satisfaction. Therefore, while the model provides insights into the individual effects of these variables, in order to find the best one requires deeper investigation.

## Predicting with a Final Model

|  |  |
| --- | --- |
| Variable | Value |
| Product Category | Household appliances |
| Market research | 17% |
| Advertising expenditure | 18% |
| Customer satisfaction | 4.3 |
| Influencer | No |
| Competitors | 3 |
| Shipping time | 2 days |

predicted\_relative\_sales = model\_polynomial\_2.predict(product\_relative\_sales)

print("Predicted Relative Sales for the product: ", round(predicted\_relative\_sales, 3))

Predicted Relative Sales for the product: 22.385

Considering the model without shipping time because of its high p-value and lack of significance in relation to relative sales, the estimated relative sales for the product are **22.385**. The principle of parsimony is applied by excluding shipping time, focusing on a more streamlined model without affecting predictive accuracy. The R-squared value, reflecting the model's explanatory power, remains at 0.82, indicating a strong fit. Trust in the model's predictions comes from the significant coefficients and their associated p-values. There might be potential limits and assumptions, but overall, the model provides a reliable estimate of relative sales.

## Non-imputed model vs. Imputed model

Models Predicting Changes in Sales

|  |  |  |
| --- | --- | --- |
|  | | |
|  | Dependent variable: relative\_sales | |
|  |  |  |
|  | Final Model | Model with imputed NaN |
|  | | |
| const | -18.33\*\*\* | -35.47\*\*\* |
|  | (0.55) | (0.69) |
| Market Research | 4.03\*\*\* | 4.66\*\*\* |
|  | (0.11) | (0.08) |
| market\_research\_sqrt | -0.16\*\*\* | -0.18\*\*\* |
|  | (0.00) | (0.00) |
| Advertising | 0.58\*\*\* | 0.61\*\*\* |
|  | (0.01) | (0.01) |
| Customer Satisfaction | 2.10\*\*\* | 2.60\*\*\* |
|  | (0.12) | (0.12) |
| Competitors | -0.28\*\*\* | -0.00 |
|  | (0.04) | (0.00) |
| Houshold products | -4.24\*\*\* | -8.44\*\*\* |
|  | (0.17) | (0.20) |
| Personal Care products | -4.00\*\*\* | -8.34\*\*\* |
|  | (0.21) | (0.22) |
| Small Electronics | -4.95\*\*\* | -9.24\*\*\* |
|  | (0.20) | (0.23) |
| Tools | -5.15\*\*\* | -9.45\*\*\* |
|  | (0.17) | (0.20) |
| Influencer No | -12.94\*\*\* | -1.71\*\*\* |
|  | (0.28) | (0.28) |
| Influencer Yes | -5.39\*\*\* | 6.00\*\*\* |
|  | (0.29) | (0.31) |
|  | | |
| Observations | 2298 | 2700 |
| R2 | 0.82 | 0.81 |
| Adjusted R2 | 0.82 | 0.81 |
| Residual Std. Error | 3.04 | 3.15 |
| F Statistic | 1022.90\*\*\* | 1152.26\*\*\* |
|  | | |
|  | \*p<0.1; \*\*p<0.05; \*\*\*p<0.01 | |
| All continuous variables have not been standardized | | |

Comparing the final model with imputed NaN values to the original final model, differences appear in the coefficients and standard errors. The inclusion of imputed NaN values has led to adjustments in the coefficients for each predictor. Specifically, the constant has shifted from -18.33 to -35.47, and some coefficients for product categories, such as Household Products, Personal Care Products, Small Electronics, and Tools, have also undergone changes. However, both models exhibit high R-squared values (0.82 in the original final model and 0.81 in the model with imputed NaN), indicating strong variance coverage by the model. The imputed model includes new coefficients for the values, influencing the overall predictive capacity. The choice between the two models should consider the impact on interpretability and alignment with the business context. In conclusion, both models provide valuable insights into the factors influencing relative sales.

## Conclusion

In conclusion, the analysis of the linear regression models yields insights into the factors influencing relative sales in a business context. The final model, incorporating market research, advertising, customer satisfaction, and competitive variables, exhibits a remarkable explanatory power with an R-squared value of 0.82. Notably, a one-unit increase in market research spending corresponds to a 4.03% increase in relative sales, underscoring the significance of strategic investment in this area, important is to mention that finding the best fit in market research is crucial since the linearity is quadratic. This means that, after a certain amount of investment relative sales start to go down, which indicates too big expenses don’t correlate with higher sales. Advertising with a one-unit increase resulted in a 0.58% boost in relative sales. Customer satisfaction emerges as a strong positive factor, indicating that a single-unit improvement corresponds to a substantial 2.10% increase in relative sales. Conversely, influencer endorsements show a significant negative impact, with a unit increase leading to a 5.39% decrease in relative sales. The exclusion of shipping time from the model for simplicity does not affect its predictive accuracy, maintaining a high R-squared value. Furthermore, the comparison between the non-imputed and imputed models highlights the robustness of findings, with both models demonstrating high predictive capacity. Overall, the analysis provides insights for strategic decision-making, emphasizing the importance of targeted market research, effective advertising, and prioritizing customer satisfaction while carefully evaluating the impact of influencer marketing.

# **Appendix**

**Code  
import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

df **=** pd**.**read\_csv('1671752sales\_results.csv')

print(df['product\_category']**.**isna()**.**sum(), df['product\_category']**.**unique())

0 ['tools' 'small\_electronics' 'household\_appliances' 'personal\_care']

*# Part 1 a.*

*#cleaning the data set from NaN values*

print(df**.**isna()**.**sum())

df\_clean **=** df**.**dropna()

print(df\_clean**.**isna()**.**sum())

*#creating dummy variables*

df\_clean **=** pd**.**get\_dummies(df\_clean, columns**=**['product\_category', 'influencer'])**.**astype(int)

df\_clean **=** pd**.**concat([df\_clean], axis**=**1)

print(df\_clean**.**describe(), df\_clean**.**nunique(), df\_clean**.**shape)

relative\_sales 0

product\_category 0

market\_research 193

advertising 70

customer\_satisfaction 0

influencer 133

competitors 0

shipping\_time 0

dtype: int64

relative\_sales 0

product\_category 0

market\_research 0

advertising 0

customer\_satisfaction 0

influencer 0

competitors 0

shipping\_time 0

dtype: int64

relative\_sales market\_research advertising customer\_satisfaction \

count 2326.000000 2326.000000 2326.000000 2326.000000

mean 5.869733 14.555460 19.613500 3.696475

std 7.101704 4.009899 4.951704 0.520426

min -38.000000 2.000000 2.000000 2.000000

25% 2.000000 12.000000 16.000000 3.000000

50% 6.000000 15.000000 20.000000 4.000000

75% 10.000000 17.000000 23.000000 4.000000

max 28.000000 28.000000 38.000000 5.000000

competitors shipping\_time product\_category\_household\_appliances \

count 2326.000000 2326.000000 2326.000000

mean -7.057610 2.168100 0.380911

std 109.528217 0.849863 0.485715

min -999.000000 1.000000 0.000000

25% 4.000000 2.000000 0.000000

50% 5.000000 2.000000 0.000000

75% 6.000000 3.000000 1.000000

max 10.000000 4.000000 1.000000

product\_category\_personal\_care product\_category\_small\_electronics \

count 2326.000000 2326.000000

mean 0.115649 0.128117

std 0.319873 0.334292

min 0.000000 0.000000

25% 0.000000 0.000000

50% 0.000000 0.000000

75% 0.000000 0.000000

max 1.000000 1.000000

product\_category\_tools influencer\_no influencer\_yes

count 2326.000000 2326.000000 2326.000000

mean 0.375322 0.792347 0.207653

std 0.484310 0.405714 0.405714

min 0.000000 0.000000 0.000000

25% 0.000000 1.000000 0.000000

50% 0.000000 1.000000 0.000000

75% 1.000000 1.000000 0.000000

max 1.000000 1.000000 1.000000 relative\_sales 56

market\_research 27

advertising 36

customer\_satisfaction 4

competitors 12

shipping\_time 4

product\_category\_household\_appliances 2

product\_category\_personal\_care 2

product\_category\_small\_electronics 2

product\_category\_tools 2

influencer\_no 2

influencer\_yes 2

dtype: int64 (2326, 12)

**import** statsmodels.api **as** sm

**import** numpy **as** np

print(df\_clean**.**describe())

*# Part 1 b.*

*#defining independent variables X (features) and defining dependent variables y (target)*

X **=** df\_clean**.**drop(columns**=**['relative\_sales'])

y **=** df\_clean['relative\_sales']

*#constant for independent variables for the intercept in the model*

X **=** sm**.**add\_constant(X)

*#fitting the model*

model **=** sm**.**OLS(y, X)**.**fit()

print(model**.**summary())

cooks\_d **=** model**.**get\_influence()**.**cooks\_distance

n **=** len(df\_clean)

df\_clean['Outlier'] **=** cooks\_d[0] **>** 4**/**n

relative\_sales market\_research advertising customer\_satisfaction \

count 2326.000000 2326.000000 2326.000000 2326.000000

mean 5.869733 14.555460 19.613500 3.696475

std 7.101704 4.009899 4.951704 0.520426

min -38.000000 2.000000 2.000000 2.000000

25% 2.000000 12.000000 16.000000 3.000000

50% 6.000000 15.000000 20.000000 4.000000

75% 10.000000 17.000000 23.000000 4.000000

max 28.000000 28.000000 38.000000 5.000000

competitors shipping\_time product\_category\_household\_appliances \

count 2326.000000 2326.000000 2326.000000

mean -7.057610 2.168100 0.380911

std 109.528217 0.849863 0.485715

min -999.000000 1.000000 0.000000

25% 4.000000 2.000000 0.000000

50% 5.000000 2.000000 0.000000

75% 6.000000 3.000000 1.000000

max 10.000000 4.000000 1.000000

product\_category\_personal\_care product\_category\_small\_electronics \

count 2326.000000 2326.000000

mean 0.115649 0.128117

std 0.319873 0.334292

min 0.000000 0.000000

25% 0.000000 0.000000

50% 0.000000 0.000000

75% 0.000000 0.000000

max 1.000000 1.000000

product\_category\_tools influencer\_no influencer\_yes

count 2326.000000 2326.000000 2326.000000

mean 0.375322 0.792347 0.207653

std 0.484310 0.405714 0.405714

min 0.000000 0.000000 0.000000

25% 0.000000 1.000000 0.000000

50% 0.000000 1.000000 0.000000

75% 1.000000 1.000000 0.000000

max 1.000000 1.000000 1.000000

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.515

Model: OLS Adj. R-squared: 0.513

Method: Least Squares F-statistic: 273.4

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:58 Log-Likelihood: -7017.8

No. Observations: 2326 AIC: 1.406e+04

Df Residuals: 2316 BIC: 1.411e+04

Df Model: 9

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -1.8968 0.544 -3.487 0.000 -2.964 -0.830

market\_research -0.5947 0.026 -23.160 0.000 -0.645 -0.544

advertising 0.5972 0.021 28.744 0.000 0.556 0.638

customer\_satisfaction 2.2182 0.198 11.216 0.000 1.830 2.606

competitors 0.0007 0.001 0.700 0.484 -0.001 0.002

shipping\_time 0.1239 0.121 1.021 0.307 -0.114 0.362

product\_category\_household\_appliances -0.1133 0.213 -0.533 0.594 -0.530 0.304

product\_category\_personal\_care 0.3811 0.279 1.367 0.172 -0.166 0.928

product\_category\_small\_electronics -1.0481 0.280 -3.748 0.000 -1.596 -0.500

product\_category\_tools -1.1166 0.214 -5.229 0.000 -1.535 -0.698

influencer\_no -4.8004 0.295 -16.280 0.000 -5.379 -4.222

influencer\_yes 2.9035 0.305 9.511 0.000 2.305 3.502

==============================================================================

Omnibus: 661.843 Durbin-Watson: 2.055

Prob(Omnibus): 0.000 Jarque-Bera (JB): 2473.602

Skew: -1.364 Prob(JB): 0.00

Kurtosis: 7.253 Cond. No. 5.75e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 8.47e-29. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

*#Detecting the ouliers using cooks d and logic (we cant have illogical numbers), at first we are checking for them.*

df\_clean**.**describe()

*#incomplete data cells*

valid\_indices **=** []

*#function to check for impossible values*

**def** check\_impossible\_values(data\_point):

**return** (

data\_point['competitors'] **>=** 0 **and**

data\_point['market\_research'] **<=** 100 **and**

1 **<=** data\_point['customer\_satisfaction'] **<=** 5 **and**

data\_point['shipping\_time'] **>** 0

)

*# Iterate through the data points*

**for** i **in** range(len(df\_clean)):

single\_data\_point **=** df\_clean**.**iloc[i]

**if** check\_impossible\_values(single\_data\_point):

valid\_indices**.**append(i)

*#create a DataFrame with data points containing valid values*

df\_clean\_valid **=** df\_clean**.**iloc[valid\_indices]

*#display the DataFrame with valid data points*

print(df\_clean\_valid, df\_clean\_valid**.**dtypes)

relative\_sales market\_research advertising customer\_satisfaction \

0 8 13 26 3

1 3 12 18 4

3 5 17 24 4

4 -5 23 18 4

5 3 17 18 4

... ... ... ... ...

2695 12 13 24 4

2696 7 14 12 4

2697 8 13 17 4

2698 8 17 20 4

2699 9 10 27 4

competitors shipping\_time product\_category\_household\_appliances \

0 5 2 0

1 5 2 0

3 7 2 1

4 4 2 0

5 4 2 1

... ... ... ...

2695 1 2 0

2696 4 3 0

2697 5 2 0

2698 6 3 0

2699 6 2 1

product\_category\_personal\_care product\_category\_small\_electronics \

0 0 0

1 0 1

3 0 0

4 0 0

5 0 0

... ... ...

2695 0 1

2696 1 0

2697 1 0

2698 0 0

2699 0 0

product\_category\_tools influencer\_no influencer\_yes Outlier

0 1 1 0 False

1 0 1 0 False

3 0 1 0 False

4 1 1 0 False

5 0 1 0 False

... ... ... ... ...

2695 0 1 0 False

2696 0 1 0 False

2697 0 1 0 False

2698 1 1 0 False

2699 0 1 0 False

[2298 rows x 13 columns] relative\_sales int64

market\_research int64

advertising int64

customer\_satisfaction int64

competitors int64

shipping\_time int64

product\_category\_household\_appliances int64

product\_category\_personal\_care int64

product\_category\_small\_electronics int64

product\_category\_tools int64

influencer\_no int64

influencer\_yes int64

Outlier bool

dtype: object

X **=** df\_clean\_valid**.**drop(columns**=**['relative\_sales'])**.**astype(float)

y **=** df\_clean\_valid['relative\_sales']

*#constant for independent variables for the intercept in the model*

X **=** sm**.**add\_constant(X)

*#fitting the model after removing impossible values*

model\_valid **=** sm**.**OLS(y, X)**.**fit()

print(model\_valid**.**summary(), df\_clean\_valid**.**isna()**.**sum())

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.663

Model: OLS Adj. R-squared: 0.661

Method: Least Squares F-statistic: 449.8

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:58 Log-Likelihood: -6512.5

No. Observations: 2298 AIC: 1.305e+04

Df Residuals: 2287 BIC: 1.311e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const 0.1801 0.488 0.369 0.712 -0.776 1.136

market\_research -0.6068 0.022 -28.190 0.000 -0.649 -0.565

advertising 0.5681 0.017 32.604 0.000 0.534 0.602

customer\_satisfaction 2.0464 0.166 12.335 0.000 1.721 2.372

competitors -0.2768 0.056 -4.921 0.000 -0.387 -0.166

shipping\_time 0.0298 0.102 0.293 0.770 -0.170 0.230

product\_category\_household\_appliances 0.2704 0.185 1.463 0.144 -0.092 0.633

product\_category\_personal\_care 0.8804 0.236 3.724 0.000 0.417 1.344

product\_category\_small\_electronics -0.2773 0.238 -1.164 0.245 -0.745 0.190

product\_category\_tools -0.6935 0.185 -3.749 0.000 -1.056 -0.331

influencer\_no -3.8137 0.262 -14.556 0.000 -4.328 -3.300

influencer\_yes 3.9938 0.270 14.788 0.000 3.464 4.523

Outlier -13.5108 0.432 -31.299 0.000 -14.357 -12.664

==============================================================================

Omnibus: 165.212 Durbin-Watson: 2.074

Prob(Omnibus): 0.000 Jarque-Bera (JB): 796.955

Skew: 0.113 Prob(JB): 8.78e-174

Kurtosis: 5.876 Cond. No. 6.99e+16

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 3.11e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular. relative\_sales 0

market\_research 0

advertising 0

customer\_satisfaction 0

competitors 0

shipping\_time 0

product\_category\_household\_appliances 0

product\_category\_personal\_care 0

product\_category\_small\_electronics 0

product\_category\_tools 0

influencer\_no 0

influencer\_yes 0

Outlier 0

dtype: int64

*#Creating a copy*

df\_clean\_no\_cooksD\_outliers **=** df\_clean\_valid**.**copy()

*#keeping rows those that are not outliers*

df\_clean\_no\_cooksD\_outliers **=** df\_clean\_no\_cooksD\_outliers[df\_clean\_no\_cooksD\_outliers['Outlier'] **==** **False**]

*#making booleans into flot numbers and splitting into dependent and independent variable*

X\_no\_cooksD\_outliers **=** df\_clean\_no\_cooksD\_outliers**.**drop(columns**=**['relative\_sales'])**.**astype(float)

y\_no\_cooksD\_outliers **=** df\_clean\_no\_cooksD\_outliers['relative\_sales']

X\_no\_cooksD\_outliers **=** sm**.**add\_constant(X\_no\_cooksD\_outliers)

*#fitting the model*

model\_no\_cooksD\_outliers **=** sm**.**OLS(y\_no\_cooksD\_outliers, X\_no\_cooksD\_outliers)**.**fit()

print("Model Summary after Removing Cook's D Outliers")

print(model\_no\_cooksD\_outliers**.**summary())

*#According to this the removing of Cook's D Outliers R-squared decreases, this mens that from this point I will be using only df\_clean\_valid data frame*

*#and we will keep the outliers which are determined by cooksD apporach*

Model Summary after Removing Cook's D Outliers

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.611

Model: OLS Adj. R-squared: 0.609

Method: Least Squares F-statistic: 382.1

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:58 Log-Likelihood: -6091.0

No. Observations: 2202 AIC: 1.220e+04

Df Residuals: 2192 BIC: 1.226e+04

Df Model: 9

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -0.0209 0.470 -0.044 0.965 -0.942 0.900

market\_research -0.5867 0.023 -25.773 0.000 -0.631 -0.542

advertising 0.5675 0.017 33.889 0.000 0.535 0.600

customer\_satisfaction 2.0083 0.159 12.649 0.000 1.697 2.320

competitors -0.2712 0.054 -5.052 0.000 -0.376 -0.166

shipping\_time 0.0597 0.097 0.613 0.540 -0.131 0.251

product\_category\_household\_appliances 0.3335 0.177 1.887 0.059 -0.013 0.680

product\_category\_personal\_care 0.6546 0.227 2.888 0.004 0.210 1.099

product\_category\_small\_electronics -0.4066 0.230 -1.768 0.077 -0.857 0.044

product\_category\_tools -0.6024 0.177 -3.394 0.001 -0.950 -0.254

influencer\_no -3.8297 0.252 -15.200 0.000 -4.324 -3.336

influencer\_yes 3.8088 0.260 14.653 0.000 3.299 4.319

Outlier 0 0 nan nan 0 0

==============================================================================

Omnibus: 35.117 Durbin-Watson: 2.054

Prob(Omnibus): 0.000 Jarque-Bera (JB): 36.542

Skew: -0.316 Prob(JB): 1.16e-08

Kurtosis: 3.003 Cond. No. inf

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 0. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

/Users/karolisliubavicius/anaconda3/lib/python3.11/site-packages/statsmodels/regression/linear\_model.py:1965: RuntimeWarning: divide by zero encountered in scalar divide

return np.sqrt(eigvals[0]/eigvals[-1])

df\_clean\_valid**.**shape, df\_clean\_no\_cooksD\_outliers**.**shape

((2298, 13), (2202, 13))

*#Part 1 c*

cont\_vars **=** [

'market\_research',

'advertising',

'customer\_satisfaction',

'relative\_sales'

]

corr\_matrix **=** df\_clean\_valid[cont\_vars]**.**corr()

plt**.**figure(figsize**=**(14, 6))

sns**.**heatmap(corr\_matrix, annot**=True**, annot\_kws**=**{'fontsize': 10}, cmap**=**'coolwarm', fmt**=**'.2f', linewidths**=**0.5)

plt**.**title("Correlation matrix of continuous variables")

plt**.**show()

*#After inspecting the correlation matrix< i can state that there is low or not significant enough correlation within continuous variables*

print(corr\_matrix)

market\_research advertising customer\_satisfaction \

market\_research 1.000000 -0.026528 0.035586

advertising -0.026528 1.000000 0.005363

customer\_satisfaction 0.035586 0.005363 1.000000

relative\_sales -0.356317 0.419391 0.144263

relative\_sales

market\_research -0.356317

advertising 0.419391

customer\_satisfaction 0.144263

relative\_sales 1.000000

*#checking for possible multicollinearity issues, the outcome there is low or almost none possibility for a multicollinearity*

**from** statsmodels.stats.outliers\_influence **import** variance\_inflation\_factor

X **=** df\_clean\_valid[['market\_research', 'advertising', 'customer\_satisfaction', 'relative\_sales']]

X **=** sm**.**add\_constant(X)

vif\_data **=** pd**.**DataFrame()

vif\_data['Variable'] **=** X**.**columns

vif\_data['VIF'] **=** [variance\_inflation\_factor(X**.**values, i) **for** i **in** range(X**.**shape[1])]

print(vif\_data)

*#check for multicolinearity, this indicates that multicollinearity doesnt exist in the model since it is < 5*

Variable VIF

0 const 79.620603

1 market\_research 1.182852

2 advertising 1.246682

3 customer\_satisfaction 1.036385

4 relative\_sales 1.468399

*#Part 1d*

independent\_variables **=** ['market\_research', 'advertising', 'customer\_satisfaction']

dependent\_variable **=** 'relative\_sales'

*#plot all continuous variables to check for linearity*

**for** var **in** independent\_variables:

sns**.**regplot(x**=**var, y**=**dependent\_variable, data**=**df\_clean\_valid, scatter\_kws**=**{"alpha":0.3})

plt**.**title(f'Regression Plot for {var} vs. {dependent\_variable}')

plt**.**tight\_layout()

plt**.**show()

*#The regeression in this case negative, this means the higher expneses or market research are the less in brings to the company*

A graph of a graph with blue dots

Description automatically generated with medium confidence

A graph of a customer satisfaction

Description automatically generated

*#apply square root transformation to 'market\_research' because the relationship seems qudratic others (competitors, advertising) are linear*

df\_clean\_valid['market\_research\_sqrt'] **=** df\_clean\_valid['market\_research'] **\*\*** 2

/var/folders/m1/102781ks0r1\_r6\_vtplwmzy00000gn/T/ipykernel\_3886/2093183772.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

df\_clean\_valid['market\_research\_sqrt'] = df\_clean\_valid['market\_research'] \*\* 2

**from** sklearn.preprocessing **import** PolynomialFeatures

*#Exploring the best fitting models*

*#checking last time for the null\_values*

print(df\_clean\_valid**.**isnull()**.**sum())

*#model from 1b linear model*

print(model\_valid**.**summary())

*#polynominal model including independent variable shipping time*

X\_polynomial **=** df\_clean\_valid**.**drop(columns**=**['relative\_sales'])**.**astype(float)

y\_polynomial **=** df\_clean\_valid['relative\_sales']

X\_polynomial **=** sm**.**add\_constant(X\_polynomial)

model\_polynomial **=** sm**.**OLS(y\_polynomial, X\_polynomial)**.**fit()

print(model\_polynomial**.**summary())

*#same polynominal with dropped shipping\_time and since it is not significant (final)*

X\_polynomial\_2 **=** df\_clean\_valid**.**drop(columns**=**['relative\_sales', 'shipping\_time'])**.**astype(float)

y\_polynomial\_2 **=** df\_clean\_valid['relative\_sales']

X\_polynomial\_2 **=** sm**.**add\_constant(X\_polynomial\_2)

model\_polynomial\_2 **=** sm**.**OLS(y\_polynomial\_2, X\_polynomial\_2)**.**fit()

print(model\_polynomial\_2**.**summary())

relative\_sales 0

market\_research 0

advertising 0

customer\_satisfaction 0

competitors 0

shipping\_time 0

product\_category\_household\_appliances 0

product\_category\_personal\_care 0

product\_category\_small\_electronics 0

product\_category\_tools 0

influencer\_no 0

influencer\_yes 0

Outlier 0

market\_research\_sqrt 0

dtype: int64

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.663

Model: OLS Adj. R-squared: 0.661

Method: Least Squares F-statistic: 449.8

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -6512.5

No. Observations: 2298 AIC: 1.305e+04

Df Residuals: 2287 BIC: 1.311e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const 0.1801 0.488 0.369 0.712 -0.776 1.136

market\_research -0.6068 0.022 -28.190 0.000 -0.649 -0.565

advertising 0.5681 0.017 32.604 0.000 0.534 0.602

customer\_satisfaction 2.0464 0.166 12.335 0.000 1.721 2.372

competitors -0.2768 0.056 -4.921 0.000 -0.387 -0.166

shipping\_time 0.0298 0.102 0.293 0.770 -0.170 0.230

product\_category\_household\_appliances 0.2704 0.185 1.463 0.144 -0.092 0.633

product\_category\_personal\_care 0.8804 0.236 3.724 0.000 0.417 1.344

product\_category\_small\_electronics -0.2773 0.238 -1.164 0.245 -0.745 0.190

product\_category\_tools -0.6935 0.185 -3.749 0.000 -1.056 -0.331

influencer\_no -3.8137 0.262 -14.556 0.000 -4.328 -3.300

influencer\_yes 3.9938 0.270 14.788 0.000 3.464 4.523

Outlier -13.5108 0.432 -31.299 0.000 -14.357 -12.664

==============================================================================

Omnibus: 165.212 Durbin-Watson: 2.074

Prob(Omnibus): 0.000 Jarque-Bera (JB): 796.955

Skew: 0.113 Prob(JB): 8.78e-174

Kurtosis: 5.876 Cond. No. 6.99e+16

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 3.11e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.817

Model: OLS Adj. R-squared: 0.816

Method: Least Squares F-statistic: 929.5

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -5809.1

No. Observations: 2298 AIC: 1.164e+04

Df Residuals: 2286 BIC: 1.171e+04

Df Model: 11

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -18.3321 0.554 -33.114 0.000 -19.418 -17.246

market\_research 4.0291 0.107 37.764 0.000 3.820 4.238

advertising 0.5838 0.013 45.477 0.000 0.559 0.609

customer\_satisfaction 2.1046 0.122 17.224 0.000 1.865 2.344

competitors -0.2775 0.041 -6.699 0.000 -0.359 -0.196

shipping\_time 0.0024 0.075 0.031 0.975 -0.145 0.149

product\_category\_household\_appliances -4.2361 0.170 -24.847 0.000 -4.570 -3.902

product\_category\_personal\_care -3.9989 0.207 -19.364 0.000 -4.404 -3.594

product\_category\_small\_electronics -4.9493 0.205 -24.124 0.000 -5.352 -4.547

product\_category\_tools -5.1477 0.170 -30.314 0.000 -5.481 -4.815

influencer\_no -12.9406 0.284 -45.644 0.000 -13.497 -12.385

influencer\_yes -5.3915 0.292 -18.473 0.000 -5.964 -4.819

Outlier -2.1401 0.410 -5.221 0.000 -2.944 -1.336

market\_research\_sqrt -0.1587 0.004 -43.939 0.000 -0.166 -0.152

==============================================================================

Omnibus: 6.171 Durbin-Watson: 2.076

Prob(Omnibus): 0.046 Jarque-Bera (JB): 6.128

Skew: 0.112 Prob(JB): 0.0467

Kurtosis: 3.118 Cond. No. 7.11e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 3.03e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.817

Model: OLS Adj. R-squared: 0.816

Method: Least Squares F-statistic: 1023.

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -5809.1

No. Observations: 2298 AIC: 1.164e+04

Df Residuals: 2287 BIC: 1.170e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -18.3294 0.547 -33.511 0.000 -19.402 -17.257

market\_research 4.0291 0.107 37.775 0.000 3.820 4.238

advertising 0.5838 0.013 45.487 0.000 0.559 0.609

customer\_satisfaction 2.1046 0.122 17.228 0.000 1.865 2.344

competitors -0.2774 0.041 -6.702 0.000 -0.359 -0.196

product\_category\_household\_appliances -4.2354 0.169 -25.027 0.000 -4.567 -3.904

product\_category\_personal\_care -3.9984 0.206 -19.436 0.000 -4.402 -3.595

product\_category\_small\_electronics -4.9484 0.203 -24.390 0.000 -5.346 -4.551

product\_category\_tools -5.1472 0.169 -30.462 0.000 -5.479 -4.816

influencer\_no -12.9391 0.280 -46.254 0.000 -13.488 -12.391

influencer\_yes -5.3903 0.289 -18.639 0.000 -5.957 -4.823

Outlier -2.1403 0.410 -5.224 0.000 -2.944 -1.337

market\_research\_sqrt -0.1587 0.004 -43.950 0.000 -0.166 -0.152

==============================================================================

Omnibus: 6.167 Durbin-Watson: 2.076

Prob(Omnibus): 0.046 Jarque-Bera (JB): 6.124

Skew: 0.112 Prob(JB): 0.0468

Kurtosis: 3.118 Cond. No. 7.28e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.9e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

*#Part 2c*

product\_relative\_sales **=** pd**.**DataFrame({

'const': [0],

'market\_research\_sqrt': [0],

'market\_research': [17],

'advertising': [18],

'customer\_satisfaction': [4.3],

'influencer\_no': [1],

'influencer\_yes': [0],

'competitors': [3],

'product\_category\_household\_appliances': [1],

'product\_category\_personal\_care': [0],

'product\_category\_small\_electronics': [0],

'product\_category\_tools': [0],

'Outlier': [0]

})

predicted\_relative\_sales **=** model\_polynomial\_2**.**predict(product\_relative\_sales)

print("Predicted Relative Sales for the product: ", round(predicted\_relative\_sales, 3))

Predicted Relative Sales for the product: 0 22.385

dtype: float64

*#Part 2d*

**from** fancyimpute **import** IterativeImputer

df\_2d **=** df**.**copy()

print(df\_2d**.**isna()**.**sum(), df\_2d**.**shape)

*#convert categorical variables to dummy variables*

df\_2d **=** pd**.**get\_dummies(df\_2d, columns**=**['product\_category', 'influencer'])

numerical\_columns **=** df\_2d**.**select\_dtypes(exclude**=**['object', 'bool'])**.**columns

*# Impute numerical variables*

imputer **=** IterativeImputer()

df\_imputed **=** pd**.**DataFrame(imputer**.**fit\_transform(df\_2d[numerical\_columns]), columns**=**numerical\_columns)

*# Display imputed DataFrame*

print(df\_imputed**.**isna()**.**sum(), df\_imputed**.**shape)

relative\_sales 0

market\_research 0

advertising 0

customer\_satisfaction 0

competitors 0

shipping\_time 0

product\_category\_household\_appliances 0

product\_category\_personal\_care 0

product\_category\_small\_electronics 0

product\_category\_tools 0

influencer\_no 0

influencer\_yes 0

dtype: int64 (2700, 12)

X **=** df\_imputed**.**drop(columns**=**['relative\_sales'])**.**astype(float)

y **=** df\_imputed['relative\_sales']

X **=** sm**.**add\_constant(X)

model\_imputed **=** sm**.**OLS(y, X)**.**fit()

print("Non-Imputed Model Summary from part 1:")

print(model\_polynomial\_2**.**summary())

print("\nImputed Model Summary:")

print(model\_imputed**.**summary())

Non-Imputed Model Summary from part 1:

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.817

Model: OLS Adj. R-squared: 0.816

Method: Least Squares F-statistic: 1023.

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -5809.1

No. Observations: 2298 AIC: 1.164e+04

Df Residuals: 2287 BIC: 1.170e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -18.3294 0.547 -33.511 0.000 -19.402 -17.257

market\_research 4.0291 0.107 37.775 0.000 3.820 4.238

advertising 0.5838 0.013 45.487 0.000 0.559 0.609

customer\_satisfaction 2.1046 0.122 17.228 0.000 1.865 2.344

competitors -0.2774 0.041 -6.702 0.000 -0.359 -0.196

product\_category\_household\_appliances -4.2354 0.169 -25.027 0.000 -4.567 -3.904

product\_category\_personal\_care -3.9984 0.206 -19.436 0.000 -4.402 -3.595

product\_category\_small\_electronics -4.9484 0.203 -24.390 0.000 -5.346 -4.551

product\_category\_tools -5.1472 0.169 -30.462 0.000 -5.479 -4.816

influencer\_no -12.9391 0.280 -46.254 0.000 -13.488 -12.391

influencer\_yes -5.3903 0.289 -18.639 0.000 -5.957 -4.823

Outlier -2.1403 0.410 -5.224 0.000 -2.944 -1.337

market\_research\_sqrt -0.1587 0.004 -43.950 0.000 -0.166 -0.152

==============================================================================

Omnibus: 6.167 Durbin-Watson: 2.076

Prob(Omnibus): 0.046 Jarque-Bera (JB): 6.124

Skew: 0.112 Prob(JB): 0.0468

Kurtosis: 3.118 Cond. No. 7.28e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.9e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

Imputed Model Summary:

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.523

Model: OLS Adj. R-squared: 0.521

Method: Least Squares F-statistic: 294.3

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -8174.5

No. Observations: 2700 AIC: 1.637e+04

Df Residuals: 2689 BIC: 1.644e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -6.1617 0.847 -7.274 0.000 -7.823 -4.501

market\_research -0.6419 0.025 -25.820 0.000 -0.691 -0.593

advertising 0.6257 0.020 31.709 0.000 0.587 0.664

customer\_satisfaction 2.6441 0.188 14.089 0.000 2.276 3.012

competitors 0.0001 0.001 0.154 0.878 -0.002 0.002

shipping\_time 0.1273 0.112 1.132 0.258 -0.093 0.348

product\_category\_household\_appliances -1.1341 0.259 -4.380 0.000 -1.642 -0.626

product\_category\_personal\_care -0.7338 0.306 -2.400 0.016 -1.333 -0.134

product\_category\_small\_electronics -2.1303 0.319 -6.686 0.000 -2.755 -1.506

product\_category\_tools -2.1636 0.262 -8.274 0.000 -2.676 -1.651

influencer\_no -1.8983 0.450 -4.217 0.000 -2.781 -1.016

influencer\_yes 5.9762 0.486 12.285 0.000 5.022 6.930

==============================================================================

Omnibus: 748.912 Durbin-Watson: 2.027

Prob(Omnibus): 0.000 Jarque-Bera (JB): 2653.681

Skew: -1.351 Prob(JB): 0.00

Kurtosis: 7.036 Cond. No. 5.36e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 1.19e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

df\_imputed['market\_research\_sqrt'] **=** df\_imputed['market\_research'] **\*\*** 2

*#creating NaN imputed model*

X\_polynomial\_2d **=** df\_imputed**.**drop(columns**=**['relative\_sales', 'shipping\_time'])**.**astype(float)

y\_polynomial\_2d **=** df\_imputed['relative\_sales']

X\_polynomial\_2d **=** sm**.**add\_constant(X\_polynomial\_2d)

model\_polynomial\_2d **=** sm**.**OLS(y\_polynomial\_2d, X\_polynomial\_2d)**.**fit()

print('Final model with no imputed values\n', model\_polynomial\_2**.**summary())

print('Model with imputed values\n', model\_polynomial\_2d**.**summary())

Final model with no imputed values

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.817

Model: OLS Adj. R-squared: 0.816

Method: Least Squares F-statistic: 1023.

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -5809.1

No. Observations: 2298 AIC: 1.164e+04

Df Residuals: 2287 BIC: 1.170e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -18.3294 0.547 -33.511 0.000 -19.402 -17.257

market\_research 4.0291 0.107 37.775 0.000 3.820 4.238

advertising 0.5838 0.013 45.487 0.000 0.559 0.609

customer\_satisfaction 2.1046 0.122 17.228 0.000 1.865 2.344

competitors -0.2774 0.041 -6.702 0.000 -0.359 -0.196

product\_category\_household\_appliances -4.2354 0.169 -25.027 0.000 -4.567 -3.904

product\_category\_personal\_care -3.9984 0.206 -19.436 0.000 -4.402 -3.595

product\_category\_small\_electronics -4.9484 0.203 -24.390 0.000 -5.346 -4.551

product\_category\_tools -5.1472 0.169 -30.462 0.000 -5.479 -4.816

influencer\_no -12.9391 0.280 -46.254 0.000 -13.488 -12.391

influencer\_yes -5.3903 0.289 -18.639 0.000 -5.957 -4.823

Outlier -2.1403 0.410 -5.224 0.000 -2.944 -1.337

market\_research\_sqrt -0.1587 0.004 -43.950 0.000 -0.166 -0.152

==============================================================================

Omnibus: 6.167 Durbin-Watson: 2.076

Prob(Omnibus): 0.046 Jarque-Bera (JB): 6.124

Skew: 0.112 Prob(JB): 0.0468

Kurtosis: 3.118 Cond. No. 7.28e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.9e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

Model with imputed values

OLS Regression Results

==============================================================================

Dep. Variable: relative\_sales R-squared: 0.811

Model: OLS Adj. R-squared: 0.810

Method: Least Squares F-statistic: 1152.

Date: Fri, 10 Nov 2023 Prob (F-statistic): 0.00

Time: 15:32:59 Log-Likelihood: -6924.9

No. Observations: 2700 AIC: 1.387e+04

Df Residuals: 2689 BIC: 1.394e+04

Df Model: 10

Covariance Type: nonrobust

=========================================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------------------

const -35.4682 0.694 -51.075 0.000 -36.830 -34.107

market\_research 4.6574 0.084 55.302 0.000 4.492 4.823

advertising 0.6119 0.012 49.257 0.000 0.588 0.636

customer\_satisfaction 2.5967 0.118 21.982 0.000 2.365 2.828

competitors -8.482e-05 0.001 -0.156 0.876 -0.001 0.001

product\_category\_household\_appliances -8.4417 0.197 -42.772 0.000 -8.829 -8.055

product\_category\_personal\_care -8.3354 0.225 -37.110 0.000 -8.776 -7.895

product\_category\_small\_electronics -9.2398 0.227 -40.724 0.000 -9.685 -8.795

product\_category\_tools -9.4513 0.199 -47.553 0.000 -9.841 -9.062

influencer\_no -1.7079 0.283 -6.026 0.000 -2.264 -1.152

influencer\_yes 5.9969 0.306 19.584 0.000 5.396 6.597

market\_research\_sqrt -0.1761 0.003 -64.030 0.000 -0.181 -0.171

==============================================================================

Omnibus: 2.209 Durbin-Watson: 2.073

Prob(Omnibus): 0.331 Jarque-Bera (JB): 2.174

Skew: -0.042 Prob(JB): 0.337

Kurtosis: 3.111 Cond. No. 9.01e+17

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.41e-28. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

**from** stargazer.stargazer **import** Stargazer, LineLocation

**from** IPython.core.display **import** HTML

*#model polynominal 2 is the final miodel without shipping time and highest r-sqaured*

Table **=** Stargazer([model, model\_valid, model\_no\_cooksD\_outliers, model\_polynomial, model\_polynomial\_2])

Table**.**title('Models Predicting Changes in Sales') *#Give it a title*

Table**.**custom\_columns(['Primary model', 'Model without illigocal values', 'Model without Cooks D outliers', 'Model with polynomial relationships', 'Final Model'], [1, 1, 1, 1, 1]) *#Give the models names*

Table**.**show\_model\_numbers(**False**) *#Remove model number*

Table**.**significant\_digits(2) *#Change decimals to 2*

Table**.**covariate\_order(['const','market\_research','market\_research\_sqrt', 'shipping\_time','advertising','customer\_satisfaction','competitors','product\_category\_household\_appliances','product\_category\_personal\_care','product\_category\_small\_electronics', 'product\_category\_tools', 'influencer\_no', 'influencer\_yes']) *#reorder variables*

Table**.**rename\_covariates({'const':'const', *#Rename relevant Variables*

'market\_research':'Market Research',

'arket\_research\_sqrt': 'Market Reserach^2',

'shipping\_time':'Shipping time',

'advertising':'Advertising',

'customer\_satisfaction':'Customer Satisfaction',

'competitors':'Competitors',

'product\_category\_household\_appliances':'Houshold products',

'product\_category\_personal\_care':'Personal Care products',

'product\_category\_small\_electronics':'Small Electronics',

'product\_category\_tools':'Tools',

'influencer\_no':'Influencer No',

'influencer\_yes':'Influencer Yes'

})

Table**.**add\_custom\_notes(['All continuous variables have not been standardized']) *#Add note about standardization*

Table**.**custom\_note\_label('') *#Remove the word "Note:"*

Table**.**show\_degrees\_of\_freedom(**False**) *#Remove the degrees of freedom*

HTML(Table**.**render\_html())

Models Predicting Changes in Sales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  | *Dependent variable: relative\_sales* | | | | |
|  |  |  |  |  |  |
|  | Primary model | Model without illigocal values | Model without Cooks D outliers | Model with polynomial relationships | Final Model |
|  | | | | | |
| const | -1.90\*\*\* | 0.18 | -0.02 | -18.33\*\*\* | -18.33\*\*\* |
|  | (0.54) | (0.49) | (0.47) | (0.55) | (0.55) |
| Market Research | -0.59\*\*\* | -0.61\*\*\* | -0.59\*\*\* | 4.03\*\*\* | 4.03\*\*\* |
|  | (0.03) | (0.02) | (0.02) | (0.11) | (0.11) |
| market\_research\_sqrt |  |  |  | -0.16\*\*\* | -0.16\*\*\* |
|  |  |  |  | (0.00) | (0.00) |
| Shipping time | 0.12 | 0.03 | 0.06 | 0.00 |  |
|  | (0.12) | (0.10) | (0.10) | (0.08) |  |
| Advertising | 0.60\*\*\* | 0.57\*\*\* | 0.57\*\*\* | 0.58\*\*\* | 0.58\*\*\* |
|  | (0.02) | (0.02) | (0.02) | (0.01) | (0.01) |
| Customer Satisfaction | 2.22\*\*\* | 2.05\*\*\* | 2.01\*\*\* | 2.10\*\*\* | 2.10\*\*\* |
|  | (0.20) | (0.17) | (0.16) | (0.12) | (0.12) |
| Competitors | 0.00 | -0.28\*\*\* | -0.27\*\*\* | -0.28\*\*\* | -0.28\*\*\* |
|  | (0.00) | (0.06) | (0.05) | (0.04) | (0.04) |
| Houshold products | -0.11 | 0.27 | 0.33\* | -4.24\*\*\* | -4.24\*\*\* |
|  | (0.21) | (0.18) | (0.18) | (0.17) | (0.17) |
| Personal Care products | 0.38 | 0.88\*\*\* | 0.65\*\*\* | -4.00\*\*\* | -4.00\*\*\* |
|  | (0.28) | (0.24) | (0.23) | (0.21) | (0.21) |
| Small Electronics | -1.05\*\*\* | -0.28 | -0.41\* | -4.95\*\*\* | -4.95\*\*\* |
|  | (0.28) | (0.24) | (0.23) | (0.21) | (0.20) |
| Tools | -1.12\*\*\* | -0.69\*\*\* | -0.60\*\*\* | -5.15\*\*\* | -5.15\*\*\* |
|  | (0.21) | (0.18) | (0.18) | (0.17) | (0.17) |
| Influencer No | -4.80\*\*\* | -3.81\*\*\* | -3.83\*\*\* | -12.94\*\*\* | -12.94\*\*\* |
|  | (0.29) | (0.26) | (0.25) | (0.28) | (0.28) |
| Influencer Yes | 2.90\*\*\* | 3.99\*\*\* | 3.81\*\*\* | -5.39\*\*\* | -5.39\*\*\* |
|  | (0.31) | (0.27) | (0.26) | (0.29) | (0.29) |
|  | | | | | |
| Observations | 2326 | 2298 | 2202 | 2298 | 2298 |
| R2 | 0.52 | 0.66 | 0.61 | 0.82 | 0.82 |
| Adjusted R2 | 0.51 | 0.66 | 0.61 | 0.82 | 0.82 |
| Residual Std. Error | 4.95 | 4.13 | 3.86 | 3.04 | 3.04 |
| F Statistic | 273.39\*\*\* | 449.84\*\*\* | 382.14\*\*\* | 929.50\*\*\* | 1022.90\*\*\* |
|  | | | | | |
|  | \*p<0.1; \*\*p<0.05; \*\*\*p<0.01 | | | | |
| All continuous variables have not been standardized | | | | | |

html\_code **=** Table**.**render\_html()

*# Specify the file path where you want to save the HTML*

file\_path **=** '/Users/karolisliubavicius/Desktop/PC files/university/Year 3/DATA modelling/deliverable/table.html'

*# Save the HTML code to the file*

**with** open(file\_path, 'w') **as** file:

file**.**write(html\_code)

print(f"Table HTML saved to {file\_path}")

Table HTML saved to /Users/karolisliubavicius/Desktop/PC files/university/Year 3/DATA modelling/deliverable/table.html

**from** stargazer.stargazer **import** Stargazer, LineLocation

**from** IPython.core.display **import** HTML

*#model polynominal 2 is the final miodel without shipping time and highest r-sqaured*

Table **=** Stargazer([model\_polynomial\_2, model\_polynomial\_2d])

Table**.**title('Models Predicting Changes in Sales') *#Give it a title*

Table**.**custom\_columns(['Final Model', 'Model with imputed NaN'], [1, 1]) *#Give the models names*

Table**.**show\_model\_numbers(**False**) *#Remove model number*

Table**.**significant\_digits(2) *#Change decimals to 2*

Table**.**covariate\_order(['const','market\_research','market\_research\_sqrt','advertising','customer\_satisfaction','competitors','product\_category\_household\_appliances','product\_category\_personal\_care','product\_category\_small\_electronics', 'product\_category\_tools', 'influencer\_no', 'influencer\_yes']) *#reorder variables*

Table**.**rename\_covariates({'const':'const', *#Rename relevant Variables*

'market\_research':'Market Research',

'arket\_research\_sqrt': 'Market Reserach^2',

'shipping\_time':'Shipping time',

'advertising':'Advertising',

'customer\_satisfaction':'Customer Satisfaction',

'competitors':'Competitors',

'product\_category\_household\_appliances':'Houshold products',

'product\_category\_personal\_care':'Personal Care products',

'product\_category\_small\_electronics':'Small Electronics',

'product\_category\_tools':'Tools',

'influencer\_no':'Influencer No',

'influencer\_yes':'Influencer Yes'

})

Table**.**add\_custom\_notes(['All continuous variables have not been standardized']) *#Add note about standardization*

Table**.**custom\_note\_label('') *#Remove the word "Note:"*

Table**.**show\_degrees\_of\_freedom(**False**) *#Remove the degrees of freedom*

HTML(Table**.**render\_html())

html\_code **=** Table**.**render\_html()**.**replace('<table', f'<table style="width:40%"')

*# Specify the file path where you want to save the HTML*

file\_path **=** '/Users/karolisliubavicius/Desktop/PC files/university/Year 3/DATA modelling/deliverable/nonimputed\_imputed.html'

*# Save the HTML code to the file*

**with** open(file\_path, 'w') **as** file:

file**.**write(html\_code)

print(f"Table HTML saved to {file\_path}")

Table HTML saved to /Users/karolisliubavicius/Desktop/PC files/university/Year 3/DATA modelling/deliverable/nonimputed\_imputed.html