PREDICTING CATALOG DEMAND REPORT

JENS GAULKE

13.01.2021

UDACITY BERTELSMANN SCHOLARSHIP

Executive Summary

Last year our company sent out its first print catalog, and is preparing to send out this year's catalog in the coming months. We have 250 new customers in our their mailing list that we want to send the catalog to. Can we predict the expected profit from this new customers? We only want to send out the catalog to these customers if the expected profit contribution exceeds $10,000.

Recommendation:

Our model predicts a revenue of **$21,987** taking into account a gross margin of 50% and deducting the printing cost of $6.5 dollars per catalog copy. This is more than the required $10,000. We should definitely send out the catalog to the new customers.

The following pages show the steps which led to the conclusion. In detail, these steps are

- Business and Data Understanding
- Analysis, Modeling, and Validation
- Presentation/Visualization

Business and Data Understanding

We need to make a decision whether or not our company should invest in a marketing campaign which includes sending catalogues out to 250 new customers. The management states that the campaign has to exceed the expected profit contribution of $10,000 to be executed. The prediction therefore needs to calculate the expected profit in order to make the decision of printing and sending the catalogues to the new customers or not.

To calculate the profit, we use the equation ***profit = revenue – cost***. To perform the calculation, we have to go some steps:

- Calculate the expected revenue from the new 250 customers considering the probability that a customer will buy if we send the catalog to him.
- Calculate the costs for the catalog and set 50% for the gross margin
- Subtract the costs from the revenue.
- If the profit is greater than $10,000, send the catalog to the new customrs

We will need this data for our prediction:

Expected profit	expected profit from catalog-induced sales – costs of printing and distributing
costs of printing and distributing	given as $6.50 per catalog
expected profit from catalog-induced sales	Expected total revenue from catalog-induced sales * average gross margin
average gross margin	Given as 50% (0.5)
expected total revenue from catalog-induced sales	expected sales volume per customer * probability of buying per customer
probability of buying per customer	given as [Score_Yes] in the dataset
expected sales volume per customer	to be predicted in the linear regression model

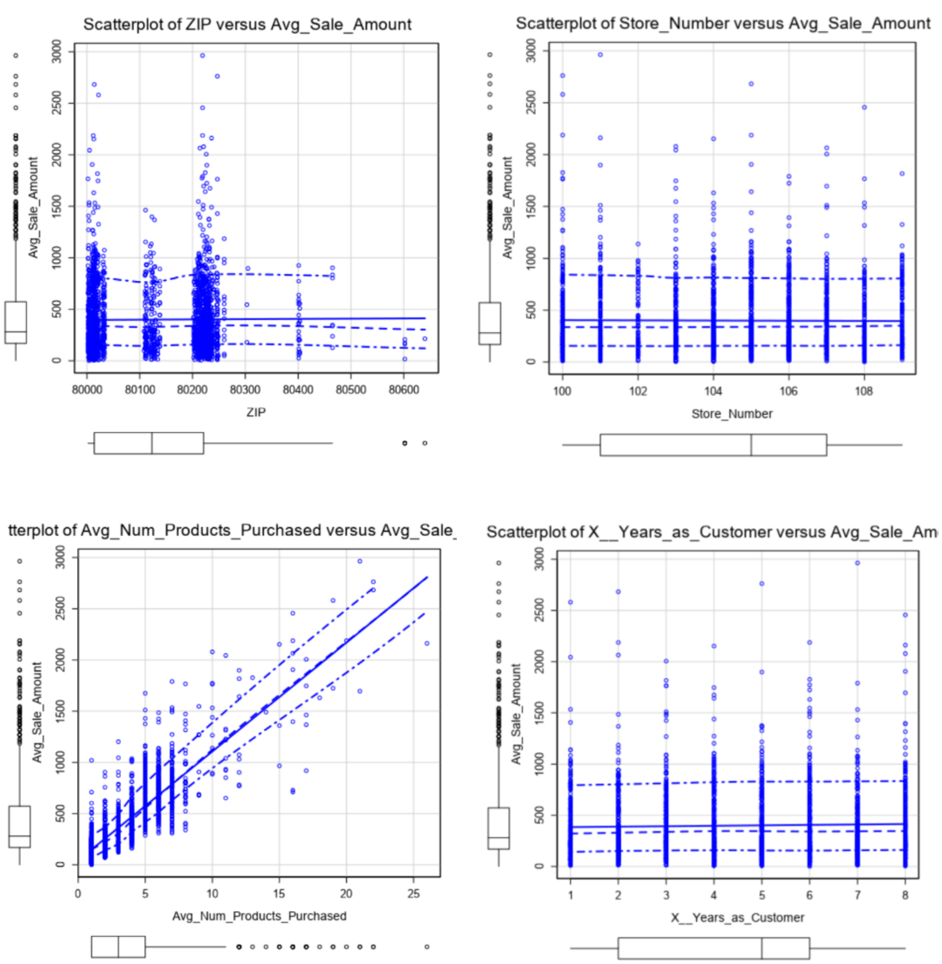
Analysis, Modeling, and Validation

We will use a linear regression model to predict the expected sales volume for each customer. Thus, the target variable (y-axis) will be **Avg_Sale_Amount**.

To make a prediction, the appropriate predictor variables from the dataset "p1-customers.xlsx" have to be chosen. The target variable **Avg_Sale_Amount** can be excluded, leaving us with eleven data columns to be considered. During the initial assessment, the following columns can directly be excluded:

- Name, Address, ID — unique by definition, so not of predictive nature
- State — all customers are from CO
- City — we have a ZIP also
- Responded_to_last_catalog — not available in new dataset used for the prediction

This leaves us with **#_Years_As_Customer**, **ZIP**, **Avg_Num_Products_Purchased** and **Store_number** as possible candidates for the linear regression. The next step was to create scatterplots to identify if a linear relationship is likely to be statistically significant.



Considering the scatterplots, we see that only for **Avg_Num_Products_Purchased** there seems to be a linear relationship. So we will assume that only for this variable a statistically significant correlation exists.

Setting up a linear regression with alteryx using **Avg_Num_Products_Purchased** gave a good result showing that our assumptions were correct. We tested the nominal/categorical variables like City and Customer_Segment as well alteryx' linear regression and found out that **Customer_Segment** is relevant for our prediction, while City is not.

Report

Report for Linear Model Linear_Regresion_Sales_Amount

Basic Summary

Call:

lm(formula = Avg_Sale_Amount ~ Customer_Segment + Avg_Num_Products_Purchased, data = the.data)

Residuals:

Min	1Q	Median	3Q	Max
-663.8	-67.3	-1.9	70.7	971.7

Koeffizienten:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	303.46	10.576	28.69	< 2.2e-16	***
Customer_SegmentLoyalty Club Only	-149.36	8.973	-16.65	< 2.2e-16	***
Customer_SegmentLoyalty Club and Credit Card	281.84	11.910	23.66	< 2.2e-16	***
Customer_SegmentStore Mailing List	-245.42	9.768	-25.13	< 2.2e-16	***
Avg_Num_Products_Purchased	66.98	1.515	44.21	< 2.2e-16	***

Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Verbleibender Standardfehler: 137.48 auf 2370 Freiheitsgrad
Mehrfach R-Quadrat: 0.8369, Angepasstes R-Quadrat: 0.8366
F-Statistik: 3040 auf 4 und 2370 Freiheitsgrad (DF), P-Wert < 2.2e-16

Type II ANOVA Analysis

Response: Avg_Sale_Amount

	Sum Sq	DF	F value	Pr(>F)	
Customer_Segment	28715078.96	3	506.4	< 2.2e-16	***
Avg_Num_Products_Purchased	36939582.5	1	1954.31	< 2.2e-16	***
Residuals	44796869.07	2370			

Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

With these result we built the model. Both predictor variables Customer_Segment as well as Avg_Num_Products_Purchased have p-values below 2.2e-16, making them statistically significant. The adjusted R-squared-value is .8366, so this can be considered a strong model.

Using the coefficients from alteryx' output, the final linear regression equation is

*Avg_Sales_Amount = 303.46 + 66.98 * (Avg_Num_Products_Purchased) – 149.36 * (Loyalty_Club_Only) + 281.84 * (Loyalty Club and Credit Card) - 245 * (Store Mailing List) + 0 * (Credit Card Only)*

Presentation/Visualization

GO!

Our Recommendation is to conduct the marketing campaign as planned as the predicted outcome is **$21,987** which is more than double the required $10,000 considered to be a success.

We tested our model with the data provided in the file p1-mailingslist.xlsx. The expected total revenue from the campaign is predicted at **$47.224** in our model.

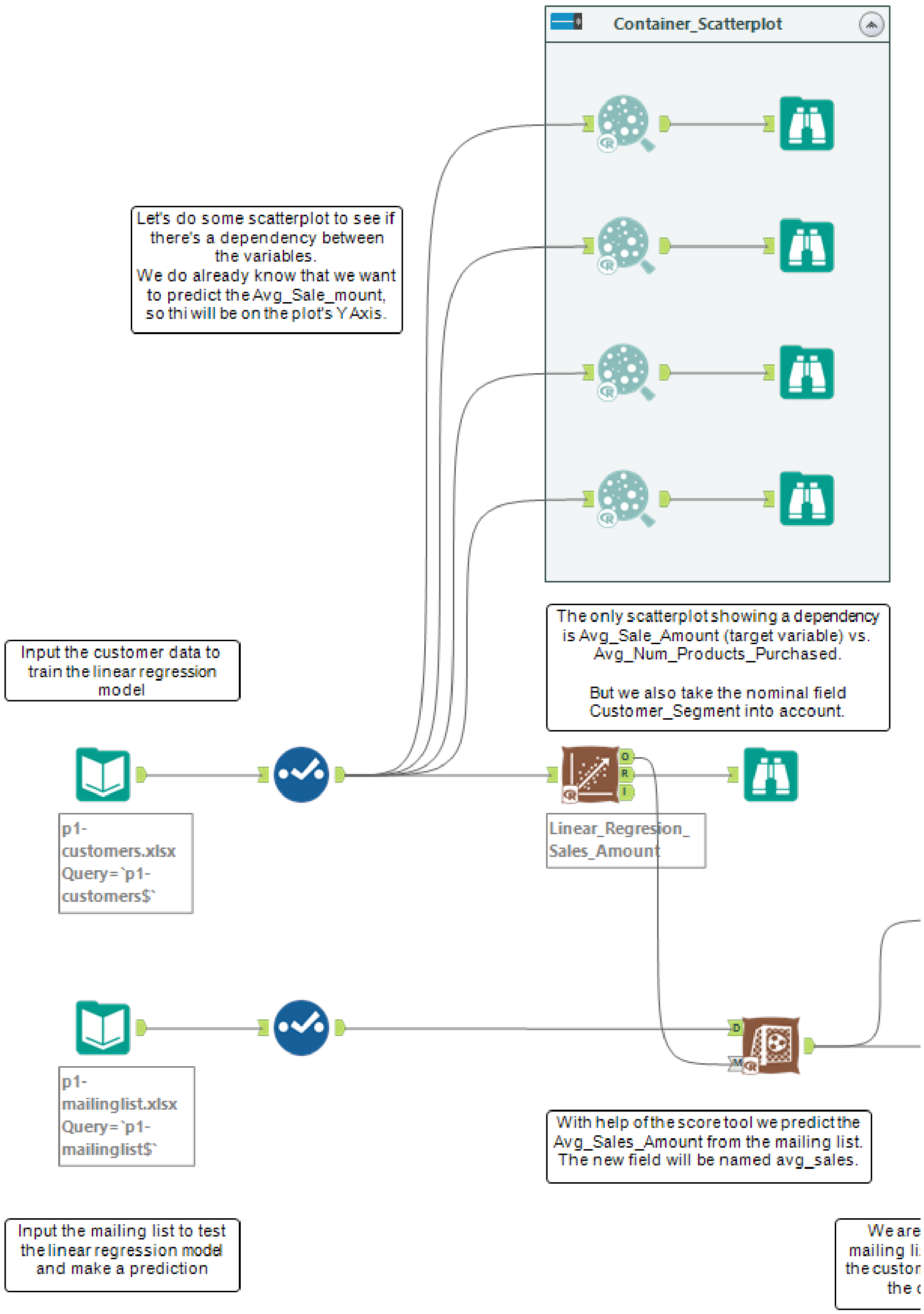
We are supposed to set an average gross margin of 50% (0.5), so the expected profit would be **$23,612**.

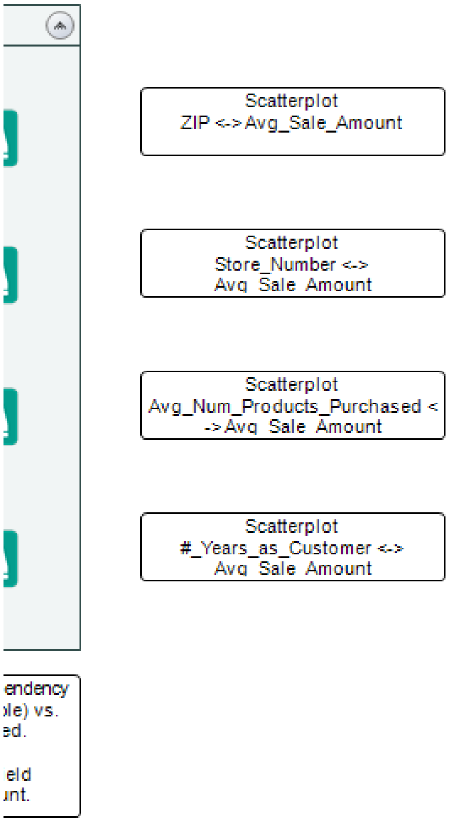
Less the cost of printing and shipping 250 catalogs to the customer which add up to 250 * $6.5 = **$1,625**, we calculate the expected profit to be **$21,987.**

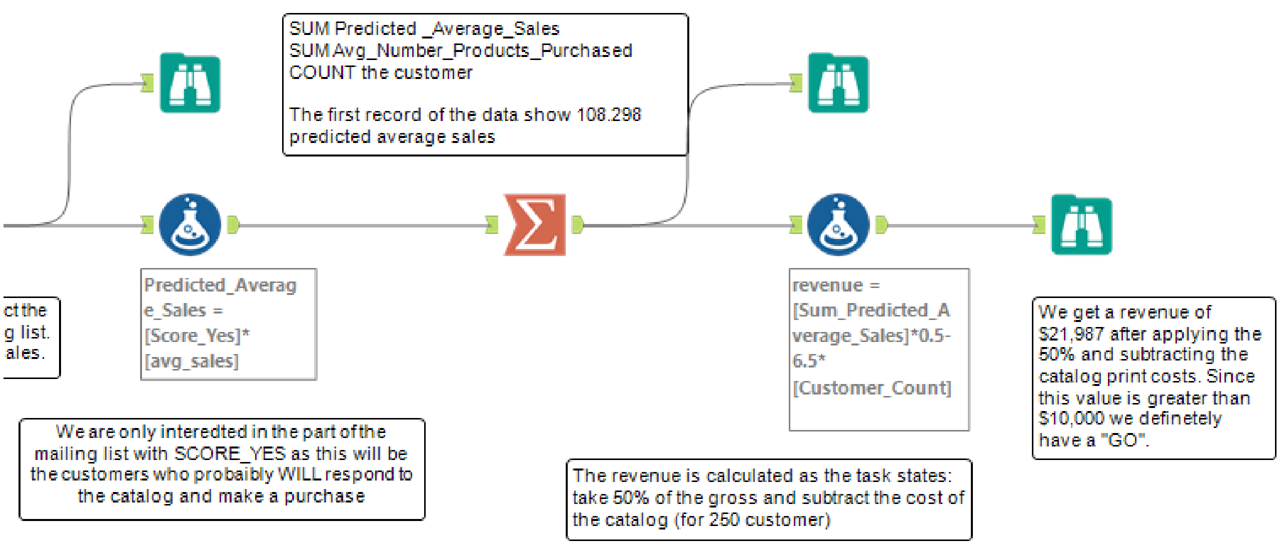
If you want to see more details on the model, please have a look at the annexes. Annex 1 shows the alteryx workflow, Annex 2 show the modeling done with Python. As you may see, both models deliver the same result

Annex 1: Alteryx workflow

We include the alteryx workflow for your convenience in this annex. It is split up in two illustrations to make it more readable. Every information needed to build your own model is in the comments in the illustrations.




Scatterplot
ZIP <-> Avg_Sale_Amount
Scatterplot
Store_Number <->
Avg_Sale_Amount
Scatterplot
Avg_Num_Products_Purchased <
-> Avg_Sale_Amount
Scatterplot
#_Years_as_Customer <->
Avg_Sale_Amount


SUM Predicted _Average_Sales
SUM Avg_Number_Products_Purchased
COUNT the customer
The first record of the data show 108.298 predicted average sales
Predicted_Average_Sales = [Score_Yes]* [avg_sales]
revenue = [Sum_Predicted_Average_Sales]*0.5-6.5* [Customer_Count]
We get a revenue of $21,987 after applying the 50% and subtracting the catalog print costs. Since this value is greater than $10,000 we definetely have a "GO".
We are only interedted in the part of the mailing list with SCORE_YES as this will be the customers who probaibly WILL respond to the catalog and make a purchase
The revenue is calculated as the task states: take 50% of the gross and subtract the cost of the catalog (for 250 customer)

Annex 2: Python / Jupyter notebook workflow

To check if the alteryx prediction is correct, we did another regression with python to prove or disprove the result. This is our workflow:

In [21]:

```
# a reference to the pandas library
import pandas as pd

# To visualize the data we need mytplotlib
# and seaborn for nice pairplots
import matplotlib.pyplot as plt
import seaborn as sns

# statsmodel will do the linear regression for us
import statsmodels.api as sm

# second library we can use for linear regression
from sklearn import linear_model
from sklearn.linear_model import LinearRegression

# the excel file must be in the same directory as this notebook
# be sure to use the right excel data file.
# This one is the customer excel file for building the model
catalog_customers_file= 'p1-customers.xlsx'

# via pandas, the contents ae read into a variable or data frame named catalog_customers_data
# pandas is able not to only read excel, but does a great job on csv, too.
catalog_customers_data = pd.read_excel(catalog_customers_file)

# now I know I can show the data by typing the variable name
catalog_customers_data
```

Out[21]:

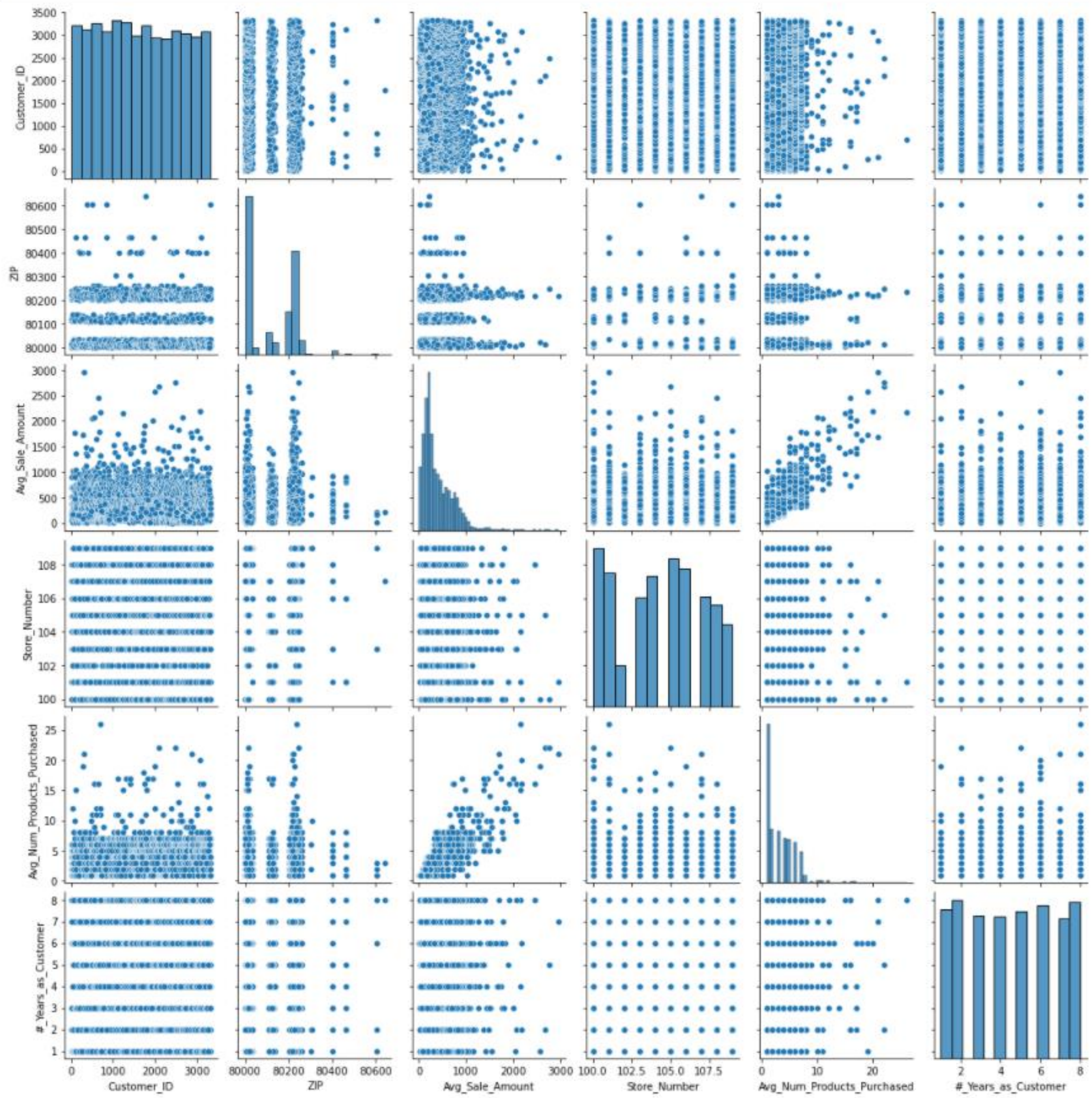
	Name	Customer_Segment	Customer_ID	Address	City	State	ZIP	Avg_Sale_Amount	Store_Number	Responded_to_Last_Catalog	Avg
0	Pamela Wright	Store Mailing List	2	376 S Jasmine St	Denver	CO	80224	227.90	100	No	
1	Danell Valdez	Store Mailing List	7	12066 E Lake Cir	Greenwood Village	CO	80111	55.00	105	Yes	
2	Jessica Rinehart	Store Mailing List	8	7225 S Gaylord St	Centennial	CO	80122	212.57	101	No	
3	Nancy Clark	Store Mailing List	9	4497 Cornish Way	Denver	CO	80239	195.31	105	Yes	
4	Andrea Brun	Store Mailing List	10	2316 E 5th Ave	Denver	CO	80206	110.55	100	Yes	
...	...	...	...	...	...	...	...	...	...	...	
2370	Joan Delisa	Credit Card Only	3287	1657 S King St	Denver	CO	80219	818.72	101	No	
2371	Helen Cordiner	Credit Card Only	3299	2102 S Lansing Ct	Aurora	CO	80014	564.93	105	No	
2372	Angela Finley	Credit Card Only	3303	1068 S Jasper St	Aurora	CO	80017	605.07	105	No	
2373	Christine Sullivan	Credit Card Only	3304	7901 W 52nd Ave	Arvada	CO	80002	656.79	107	No	
2374	Elissa Engledow	Credit Card Only	3315	9360 E Center Ave	Denver	CO	80247	167.59	104	No	

2375 rows × 12 columns

In [7]:

```
# with a seaborn pairplot, we can check for linear relationship
# in each and every combination of the data columns.
# as we can see, the only relationship seems to be between
# Avg_Sales_Amount and Avg_Num_Products_Purchased

sns.pairplot(catalog_customers_data)
plt.show()
```



In [73]:

```
# We drop the columns without statistical significance from our dataset
cleaned_customers_data = catalog_customers_data.drop(['Customer_ID', 'ZIP', 'Store_Number', '#_Years_as_Customer'], axis=1)
#we also can drop the nominal data without significance as described in analysis section
cleaned_customers_data = cleaned_customers_data.drop(['Name', 'Address', 'City', 'State', 'Responded_to_Last_Catalog'], axis=

# we still have nominal data in the dataset, namely Customer_Segment
# As this column is relevant for us, we have to create dummy variables
# and we have to drop "Credit card only" as stated in the task description
cleaned_customers_data = pd.get_dummies(cleaned_customers_data, columns=['Customer_Segment'], drop_first=True)

# set the target variable
Y = cleaned_customers_data['Avg_Sale_Amount']
# set the predictor variables
X = cleaned_customers_data.drop(['Avg_Sale_Amount'], axis=1)

# let's to the evaluation with statsmodels
# we have to add a constant to the calculation or
# we do not have a Y-intercept
X = sm.add_constant(X)

# build the model
model = sm.OLS(Y,X).fit()
model_prediction = model.predict(X)
model_details = model.summary()

# print the details, so we can compare to alteryx
model_details

X
```

Out[73]:

	const	Avg_Num_Products_Purchased	Customer_Segment_Loyalty Club Only	Customer_Segment_Loyalty Club and Credit Card	Customer_Segment_Store Mailing List
0	1.0	1	0	0	1
1	1.0	1	0	0	1
2	1.0	1	0	0	1
3	1.0	1	0	0	1
4	1.0	1	0	0	1
...	...	...	...	...	...
2370	1.0	5	0	0	0
2371	1.0	6	0	0	0
2372	1.0	6	0	0	0
2373	1.0	7	0	0	0
2374	1.0	1	0	0	0

2375 rows × 5 columns

In [85]:

```
# Now we have a working model, we can use the data from
# p1-mailinglist.xlsx to test the model and to make a prediction
catalog_test_file = "p1-mailinglist.xlsx"
catalog_test_data = pd.read_excel(catalog_test_file)

# see above
cleaned_test_data = pd.get_dummies(catalog_test_data, columns=['Customer_Segment'], drop_first=True)
cleaned_test_data = cleaned_test_data.drop(['Customer_ID', 'ZIP', 'Store_Number', '#_Years_as_Customer'], axis=1)
cleaned_test_data = cleaned_test_data.drop(['Name', 'Address', 'City', 'State'], axis=1)

# even if we know we need Score_Yes amd Score_No. these columns are not present in the customer dataset
# so a correlation between the model data and the test data will not work
# alteryx can do this automatically, we have to get our hands dirty
cleaned_test_data1 = cleaned_test_data.drop(['Score_Yes', 'Score_No'], axis=1)

# set the predictor variables
X_test = cleaned_test_data1

# let's to the evaluation with statsmodels
# we have to add a constant to the calculation or
# we do not have a Y-intercept
X_test = sm.add_constant(X_test)


# and make a prediction with the test data
model_prediction_test = model.predict(X_test)

# we have to make a prediction according to the value of Score_Yes
# so let's read the score values again and add them to the model
score = pd.read_excel('p1-mailinglist.xlsx', usecols=['Score_Yes'])

# add the model to the data frame
score['Predicted_Sales'] = model_prediction_test

# show the results
score
```

Out[85]:

	Score_Yes	Predicted_Sales
0	0.305036	355.036364
1	0.472725	987.159466
2	0.578882	622.941184
3	0.305138	288.060159
4	0.387706	422.012569
...	...	...
245	0.216194	1509.035160
246	0.192800	355.036364
247	0.423456	555.964979
248	0.259251	772.296906
249	0.203650	638.344496

250 rows × 2 columns

In [108]:
```
# now we do the evaluation

print("Catalog marketing campaign results")
print("---------------------------------\n")

total_revenue = sum(score['Predicted_Sales']*score['Score_Yes'])
print("Total expected revenue from the marketing campaign: ", '${:,.2f}'.format(total_revenue))

adjusted_revenue = total_revenue * 0.5
print("Applying gross margin of 50%: ", '${:,.2f}'.format(adjusted_revenue))

number_customers = score.shape[0]
print_costs = 6.5
catalog_costs = print_costs * number_customers
print("Print costs are $6.5 x", number_customers, "customer(s):", '${:,.2f}'.format(catalog_costs))

print("")
expected_profit = adjusted_revenue - catalog_costs
print("The expected profit from the marketing campaign is:", '${:,.2f}'.format(expected_profit))

if (expected_profit > 10000):
    print("Recommendation: GO! We should do the campaign.")
else:
    print("Recommendation: NO GO! We should not do the campaign.")
```

```
Catalog marketing campaign results
---------------------------------

Total expected revenue from the marketing campaign:  $47,224.87
Applying gross margin of 50%:  $23,612.44
Print costs are $6.5 x 250 customer(s): $1,625.00

The expected profit from the marketing campaign is: $21,987.44
Recommendation: GO! We should do the campaign.
```

As expected, Python computed the same results as alteryx.