INSIDE

Competitive Pricing

Channel Pricing

Margin Waterfall

Price Modeling

Value Pricing

PRICING PROFITABLY IN A COMPETITIVE MARKET WITHOUT SELLING ON PRICE

PROJECT REVIEW

AUTHOR: BRENT JANAKY, MBA

janakybrent@gmail.com

Table of Contents

WELCOME

Hi, I'm Brent and I'm excited to share my real-world pricing story with you.

I created this report to help you and others understand how I conduct business and deliver value. And maybe you'll learn a thing or two about pricing as you follow along.

I'll be specifically addressing how to profitably price products in competitive environments. This is based on my real-world experience in the cannabis industry (specifically, hemp).

We'll dig into the competition's pricing, cost of goods sold (COGS), margin waterfalls, gross margins, MSRPs, pricing for distributors, wholesalers and e-commerce (direct-to-consumer).

About Me:
I enjoy tackling the tough business challenges that really move the needle. After all, its measurable, quantifiable results that matter.

I've worked in various technology industries for over 20 years. I have a natural tendency to operationalize things. As a scientist with an MBA, I'm inherently curious about why business problems exist and why they haven't been solved yet. And given that I really enjoy problem solving, I'm considered a "fixer".

I am a data addict. I have a great respect for data and analytics and how they can be leveraged into a sustainable competitive advantage. This includes pricing strategy as it is heavily reliant on data (internal and external).

I enjoy working with diverse teams to uncover answers to those pressing questions related to growth and profitability. How do we grow, sustainably? What products should we sell? How should we price a product when there are a lot of other options? How do we keep customers buying from us? Why do customers only seem to care about our price? And so many other questions!

“If you can’t measure it, you can’t improve it.” - Peter Drucker

"Price is what you pay. Value is what you get." - Warren Buffett

“Perhaps the reason price is all your customers care about is because you haven’t given them anything else to care about.”– Seth Godin

EXECUTIVE SUMMARY

This project took place in a startup company where I was one of the managing partners. We operated in the hemp arm of the cannabis industry where we developed, manufactured and sold hemp-based wellness products.

The pricing project was undertaken so we could price our products in such a way that all partner channels could be profitable and we could get a foothold in the industry.

We sold direct-to-consumers (DTC) via our e-commerce website, to wholesalers (retail stores) and to distributors. Everyone needs to be profitable for business to take place. Furthermore, we needed to sell beyond price to avoid commoditizing our products. This is where selling on the product benefits takes over.

Since there are few barriers to entry, the hemp (CBD) industry is incredibly competitive with thousands of brands. It's easy to fall prey to a cost-plus margin pricing approach as it's seemingly easy. But that could lead to prices that leave a lot of profit on the table, poor sales or a myriad of other problems.

By combining competitive pricing intelligence with a solid understanding of all of the costs that went into our products, I was able to develop product pricing that resonated with our target market. I was also able to develop pricing that worked for e-commerce sales (direct-to-consumer), distributors and wholesalers.

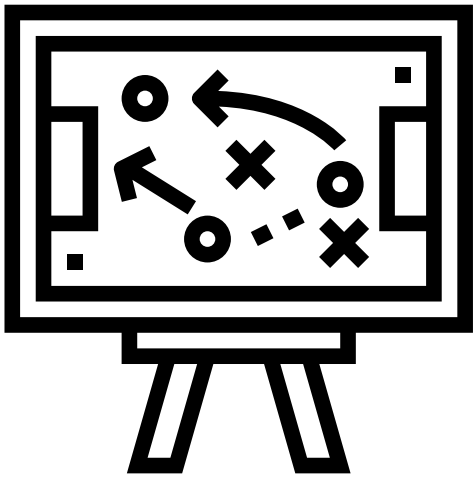
To simplify the purchase process and increase customer tenure, I launched a subscription ('autoship') pricing program. It offered a significant discount to customers that signed up for the program. Basically, customers would sign-up for a monthly shipment of product and it would show up automatically without them needing to place repeat orders.

I also used various promotions and a loyalty points program to further enhance the average customer tenure which drove customer lifetime value.

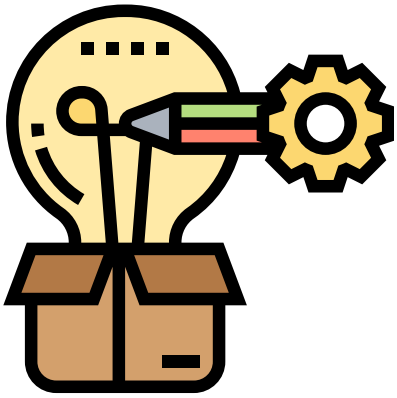
Finally, by focusing on the benefits of our products versus technical features, I was able to drive a price premium as our value proposition and messaging could better align with our target market (which was intentionally quite targeted).

MY APPROACH

1. Develop products with benefits that a target market actually cares about and will pay for.
2. Determine who the largest competitors are.
3. Gather competitive data, including pricing, & build a database.
4. Develop a linear regression of competitive prices (DTC and wholesale) as a function of quantities. Use derived formulas to develop initial price points.
5. Determine cost of goods sold (COGS) for each product SKU.
6. Using calculated (regression) prices and COGS, develop margin waterfalls.
7. Develop a promotional plan with discounts & rerun waterfall.
8. Develop loyalty point program & rerun waterfall.
9. Develop subscription ("autoship") program & rerun waterfall.
10. Evaluate worst-case scenario for discounts (maximum discounting per order) and determine gross margin.
    a. Example: DTC subscription order with loyalty points and a promotional coupon and free shipping.
11. Develop distributor and wholesale prices and calculate gross margin for each channel.
12. Ensure prices make sense by SKU and channel and ensure no channel price conflicts.



THINK BENEFITS BEFORE PRICE



Before I can jump into pricing, I want to briefly mention product development. At the end of the day, regardless of how well priced I think my products are, if nobody cares about them or understands how they will help solve their problems, I won't get a penny.

The Absolute Basics:

1. Pick a target market you can reach. Basic Example: Busy women between the ages of 30-40 in management-level positions that are interested in continuing education to help them reach the next level in their careers.
2. Understand what motivates them, fears, dreams, etc.
3. Develop products, pricing & messaging that address these ladies' wants and desires.
4. Create a sales & marketing strategy to reach those potential buyers.

Now that I have a great product to sell and a value proposition that resonates with a specific target market I can reach, it's time to start selling the benefits of the product. I do not simply sell on price. That's a losing proposition in the long run. If I only sold on price, I would clearly not be selling based on the product benefits and my target market would not be compelled to buy based on the value my products deliver.

The customers that buy based on price are what I call "price hunters". They don't care about anything other than price. They are not loyal and will jump ship as soon as they find the next lowest price. Who wants that type of customer? No me. I'm after loyal customers that are willing to pay a premium for my product because it actually solves one of their pressing problems. They will trust my brand and continue to purchase from me, keeping my customer acquisition costs down and will evangelize my brand. What a win!

COMPETITIVE PRICING

One of the first things I do when pricing products is to see what the competition is doing. I gravitate to what the more successful, large players are doing. This gives me a benchmark to start with. Later I'll discuss how my products' differentiators can add additional value, thus price.

I will research the competition and build a database of their products, features, value propositions and pricing. I do this for retail (DTC) and wholesale prices. From there, I can create an Excel model which is a function of their products' pricing and quantity of active ingredient (in this case, CBD). I also like to use Python to perform analytics. I'll show the output of that as well.

Below is a simple example using fictitious data. I search the Internet to find competitive pricing data. Here, I'm looking at different size bottles of CBD oil which is defined by Milligrams CBD. The column "Price/Milligram" is a calculated column which I may or may not use.

My Product: Oil 2500 (2,500mg CBD per Bottle)

Major Competitor	Product	E-commerce Price	Milligrams CBD	Price/Milligram
A	1	$42.00	250	$0.168
G	2	$49.00	250	$0.196
C	3	$100.00	500	$0.200
G	4	$120.00	750	$0.160
C	5	$120.00	1000	$0.120
D	6	$130.00	1000	$0.130
A	7	$148.00	1500	$0.099
B	8	$142.00	1500	$0.095
C	9	$158.00	1900	$0.083
D	10	$160.00	2000	$0.080
A	11	$175.00	2200	$0.080
B	12	$180.00	2300	$0.078
D	13	$175.00	2300	$0.076
C	14	$193.00	2500	$0.077
E	15	$180.00	2500	$0.072
F	16	$190.00	2600	$0.073
G	17	$200.00	2800	$0.071
D	18	$200.00	3000	$0.067

CREATE A PRICE MODEL CONT'D.

Hint:
I can create a model that will predict prices for a variety of products simply by adding in more price points for other products at different sizes (milligrams). As seen in the data in the previous page, I have pricing for small bottles (250mg CBD) all the way up to large 3,000mg CBD bottles.

In our company, we had a variety of products (in the same product category) to price so why not create a parent curve that will predict prices for an entire product line. That can really help with keeping your prices aligned.

Keep in mind, these predicted prices are only starting points for our pricing. I will adjust our final prices to account for value-add benefits and features of our products. For example, our products are made in a GMP-grade cleanroom facility (the gold standard for quality in the industry). I could possibly charge more for this feature if I could translate this into a benefit that my target market appreciated.

> “..why not create a parent curve that will predict prices for an entire product line”

CREATE A PRICE MODEL CONT'D.

Using the competitive data in my database, I can create a model which predicts a retail sales price for a given bottle size (i.e., 2500mg CBD bottle). Some times a simple linear model in the form y = m*x + b will work but I don't always expect that. I fit multiple models to the data to find a really good fit (r-squared value) and sometimes a non-linear model fits much better. Let's take a look at what I mean.

Let's start with what this looks like in Excel and then I'll show how I do this in Python. The results are the same whether I use Excel or Python.

Below you can see that by using the competitive price points, I can create a scatter plot and use Excel's 'trendline' function to fit a linear regression line with a decent fit with the formula: y = 0.0506x+61.084 (r-squared = 0.9192). I will use this equation to generate a predicted price for my product which is generically called "Oil 2500" (2,500 milligrams CBD).

Predicted Price for 2500mg bottle: $187.70 (linear model)

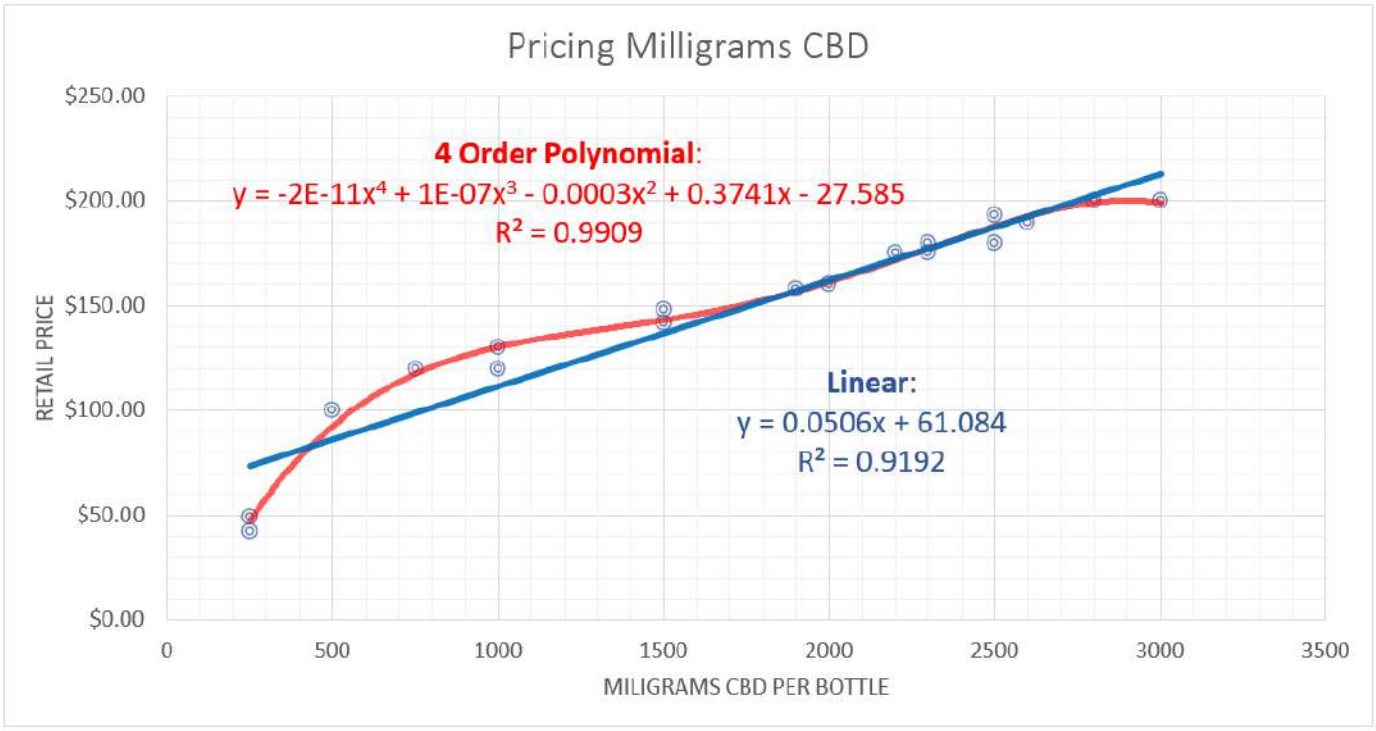
I will try to fit different models to the data to find a better fit vs. the line fitted above. In this case, I found that a 4 order polynomial had a really excellent fit with an r-squared = 0.9909 and the resulting formula: y = -2E-11x4 + 1E-07x3 - 0.0003x2 + 0.3741x - 27.585.

Predicted Price for 2500mg bottle: $187.91 (4 order polynomial model)

What I noticed was that the predictions between the two models were nearly identical but when I used the models to predict prices at the low end (around 250 mg), the polynomial model performed much better.

Let's take a look at the scatter plot and resulting models.

CREATE A PRICE MODEL CONT'D.



As you can see, both models fit very well from around 2000 milligrams to 2800 milligrams. But, outside of that range the linear model underperforms compared to the polynomial model. That is why I try different models and I evaluate them with different inputs that are represented by my product line.

As an example, if we had a product that was Oil 250 (250 milligrams CBD), the linear model predicted a price of $73.75 while the polynomial model predicted $47.36 which is much more reasonable compared to the actual competitive prices.

CREATE A PRICE MODEL CONT'D.

Excel Caution:

Excel's trendline function can produce an equation for the fitted trendline (as seen in my chart). I do **not** recommend using that equation as an input to a cell. Due to Excel rounding numbers and polynomial functions potentially having very large numbers, you can end up with nonsensical outputs.

If using Excel's trendline equations, understand the LINEST function which fits X, Y data to a polynomial without the rounding.

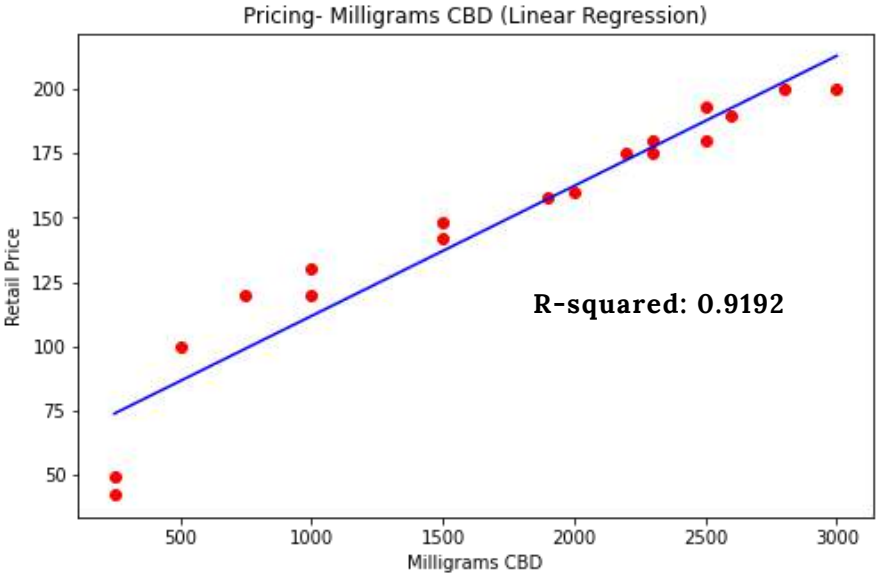
I often prefer running models in Python for a variety of reasons. If you're familiar with Python, great, let's take a look at how to run regression models on our data. You can find my GitHub files here. You'll notice in my GitHub I've created a .csv file containing only the pricing data. The X (independent variable) values are in column 0 and the y (dependent variable) values are in column 1.

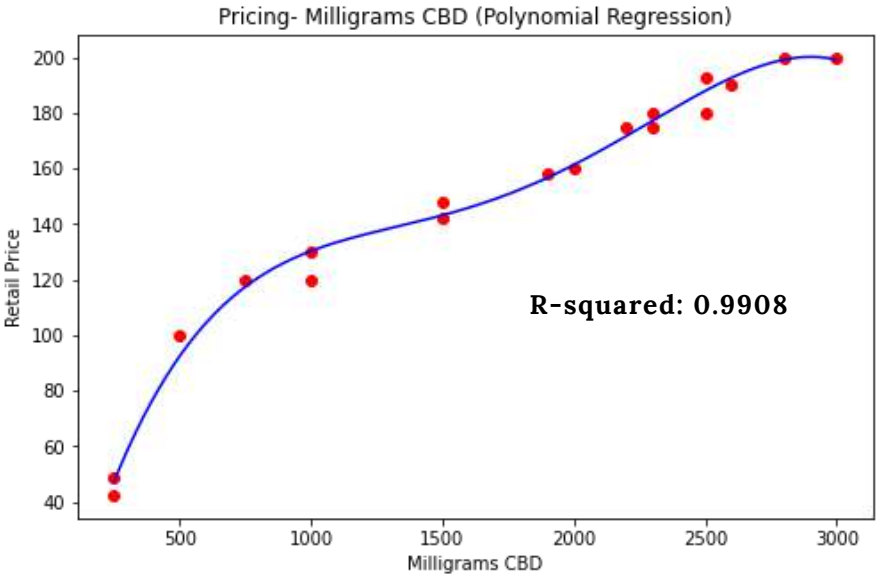
Pages 16-19 contains screenshots of the Python (Jupyter notebook) code for convenience.

Note: Thankfully, Python does not have the same 'rounding' headache as Excel.

With only a few libraries (Numpy, Matpltolib and Pandas), we can generate linear and polynomial models with their corresponding R-squared values. These are the same regression models that I built in Excel. Let's take a look at these.

CREATE A PRICE MODEL CONT'D.

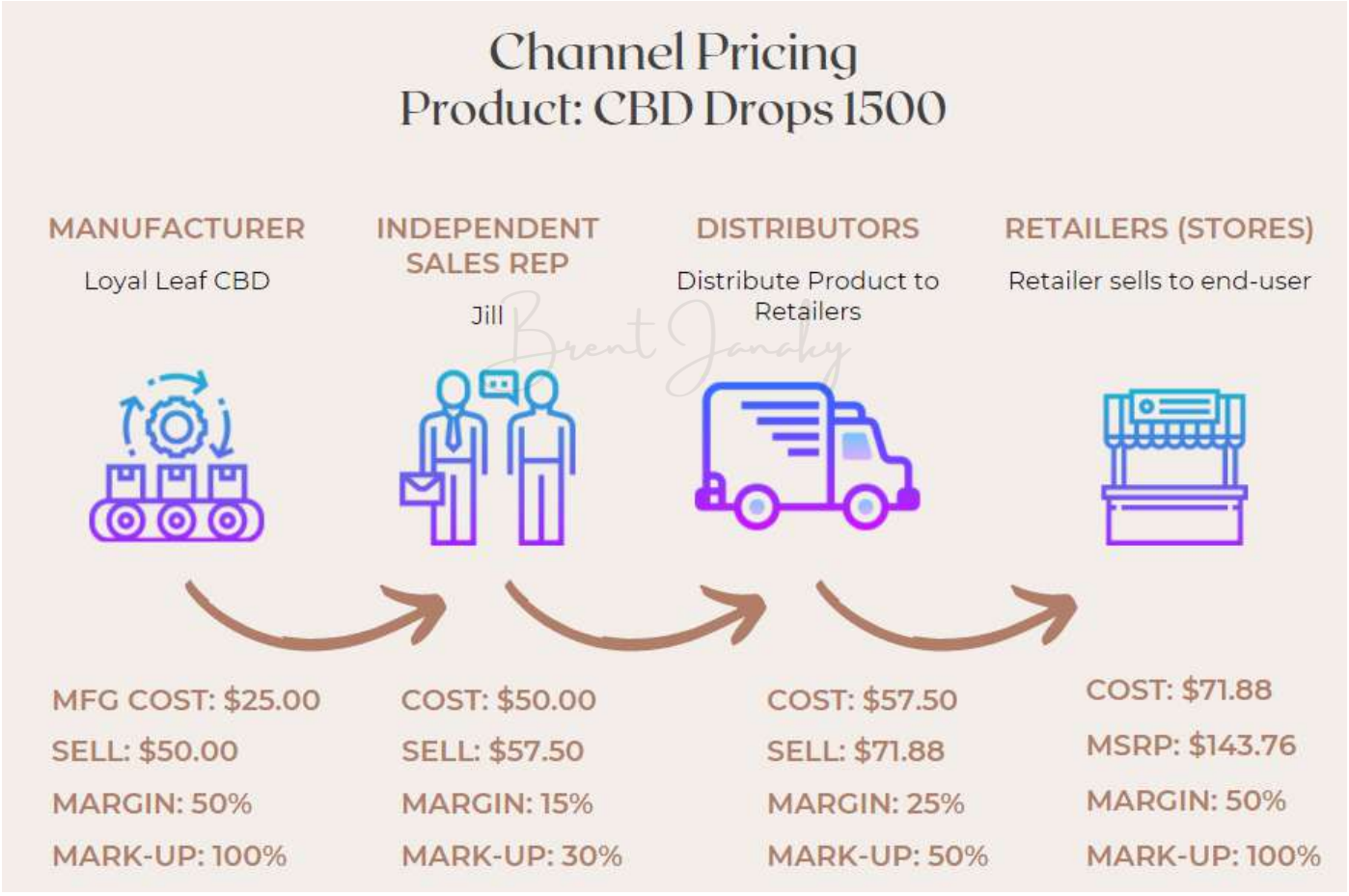




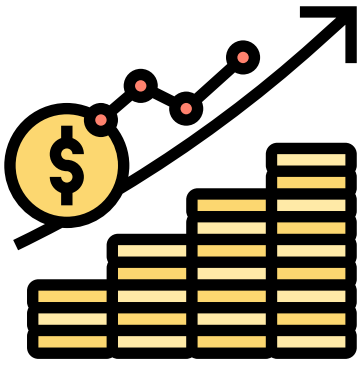
CHANNEL PRICING

Pricing can get tricky as you add more players into the mix. I had to create pricing for DTC (direct to consumers through our e-commerce website), distributors (they sell to retailers) and wholesalers (retail stores). I also had to take into account sales reps and their commissions. There was a lot to take into account to make sure all the channels received profitable pricing.

Let's take a look at an example of what this looks like using some fictitious data. You can see that if our cost is $25 and we sold DTC at the MSRP of $143.76, our gross margin would be around 87%!



CHANNEL PRICING AND WATERFALL



I don't believe in the "I think our gross margin is too high" mentality when pricing. Often times it's a necessity due to all the down-stream channel pricing and mark-ups that have to take place. Furthermore, I believe in market pricing versus cost-plus profit margin pricing. If our gross margins are really, really high and we're honing in on market prices, that's great! We have our costs really under control and we've earned those gross margins.

To help keep an eye on our profitability, I use a tool called a Pock Margin Waterfall. It's a simple Excel chart that helps visualize where our relevant costs are coming from. It helps show the profit impact of a promotion or special discount amidst all of the other costs that impact our gross margin.

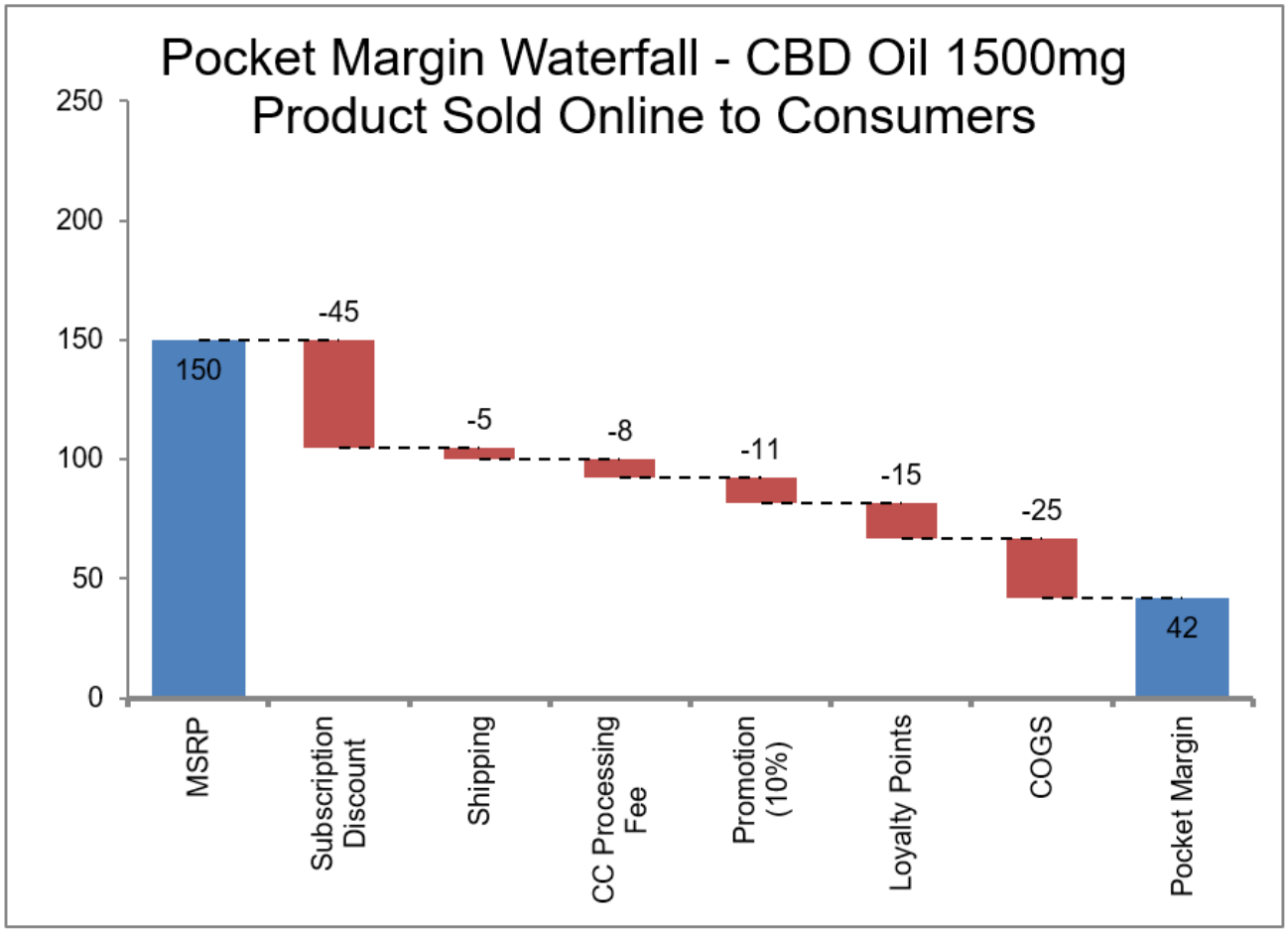
For this example, I'm evaluating our DTC pocket margin waterfall. What you'll see is that I start with our retail (MSRP) price of $150.00 and then subtract relevant **costs**. These costs are subscription program discounts, shipping costs (that we absorb), credit card processing fees, promotional discounts, loyalty point discounts/credits, and cost of goods sold (COGS).

Pocket Margin = Selling Price - Sum(costs)

I will do this for each channel as each is unique. They will have various pricing, payment terms, discounts, shipping fees, etc.

Let's take a look at a fictitious example on the next page. The numbers are different than what was previously shown on the channel pricing visualization. This is just for illustrative purposes.

POCKET MARGIN EXAMPLE



SELLING THE BENEFITS

At this point, I've done a lot of work and my coffee maker is struggling to keep up. I have my competitive price and products database and price models with starting prices. Now I can evaluate my products' benefits and features against the competition's and see where my product pricing needs to fit.

Example:
My Oil 2500 (2500 milligrams CBD) product has additional botanical ingredients to enhance sleep and amino acids to improve exercise recovery.

It's branded to appeal to our target market and the messaging sings to their desires. The price model is predicting a price of $187.91 but I believe I can get a 20% lift for the value my product delivers. That puts the MSRP at $225.49. I'll start here and see what the response is. I have plenty of gross margin to toggle price via discounting to get an idea of price elasticity.

PYTHON CODE

Predicting Hemp Oil Prices Using Regression

Importing the libraries

```
[ ]  1 import numpy as np
     2 import matplotlib.pyplot as plt
     3 import pandas as pd
```

Importing the dataset

```
[ ]  1 dataset = pd.read_csv('Pricing.csv')
     2 X = dataset.iloc[:, 0:-1].values
     3 y = dataset.iloc[:, -1].values
```

Training the Linear Regression model on the whole dataset

```
[ ]  1 from sklearn.linear_model import LinearRegression
     2 lin_reg = LinearRegression()
     3 lin_reg.fit(X, y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Training the Polynomial Regression model on the whole dataset

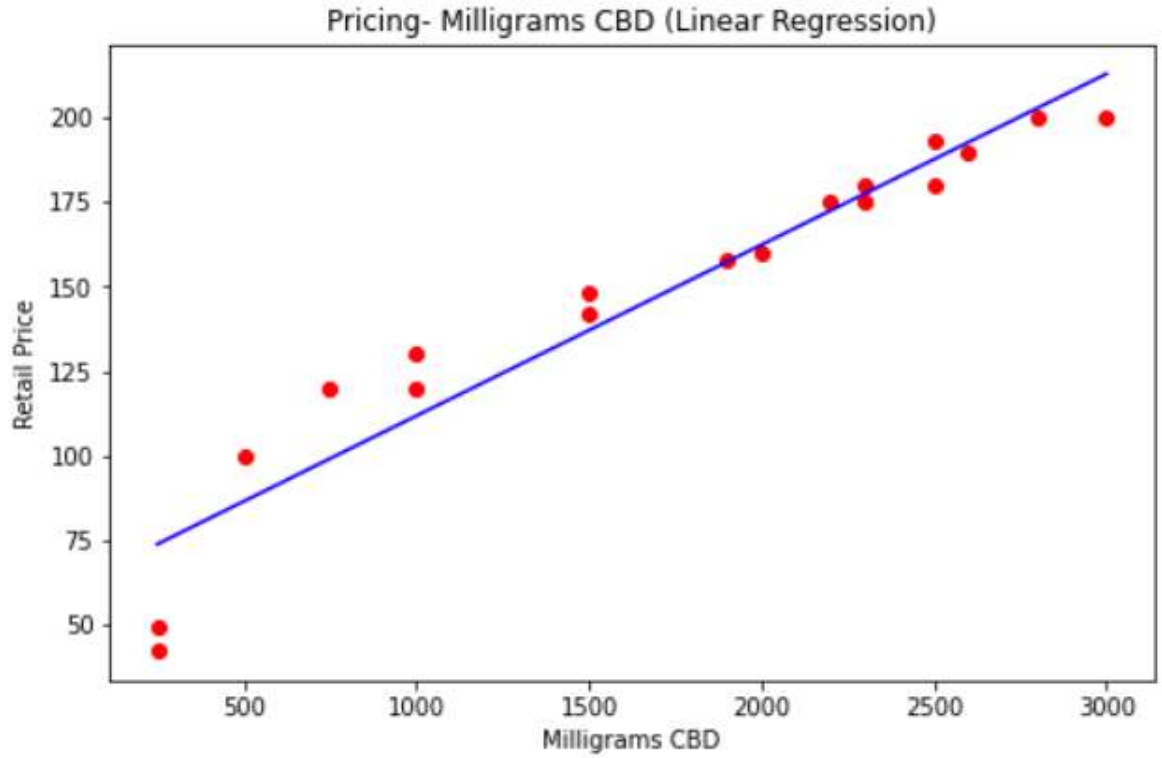
```
[ ]  1 from sklearn.preprocessing import PolynomialFeatures
     2 poly_reg = PolynomialFeatures(degree = 4)
     3 X_poly = poly_reg.fit_transform(X)
     4 lin_reg_2 = LinearRegression()
     5 lin_reg_2.fit(X_poly, y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

PYTHON CODE

Visualizing the Linear Regression results

```
[ ]  1 plt.figure(figsize=(8,5))
     2 plt.scatter(X, y, color = 'red')
     3 plt.plot(X, lin_reg.predict(X), color = 'blue')
     4 plt.title('Pricing- Milligrams CBD (Linear Regression)')
     5 plt.xlabel('Milligrams CBD')
     6 plt.ylabel('Retail Price')
     7 plt.show()
     8 from sklearn.metrics import r2_score
     9 print('r-squared:', r2_score(y, lin_reg.predict(X)))
```

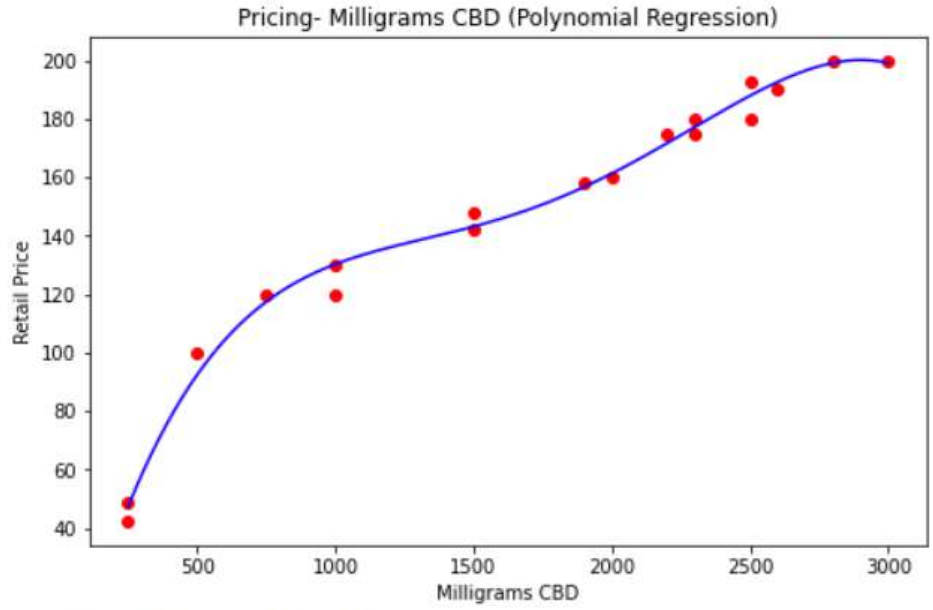



```
r-squared: 0.9192135813924626
```

PYTHON CODE

Visualizing the Polynomial Regression results

```
plt.figure(figsize=(8,5))
X_grid = np.arange(min(X), max(X), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
plt.title('Pricing- Milligrams CBD (Polynomial Regression)')
plt.xlabel('Milligrams CBD')
plt.ylabel('Retail Price')
plt.show()
from sklearn.metrics import r2_score
print('r-squared:', r2_score(y, lin_reg_2.predict(poly_reg.fit_transform(X))))
```




```
r-squared: 0.9908541086117513
```

PYTHON CODE

Predicting a new result with Linear Regression

```
[ ]  1 lin_reg.predict([[2500]])
```

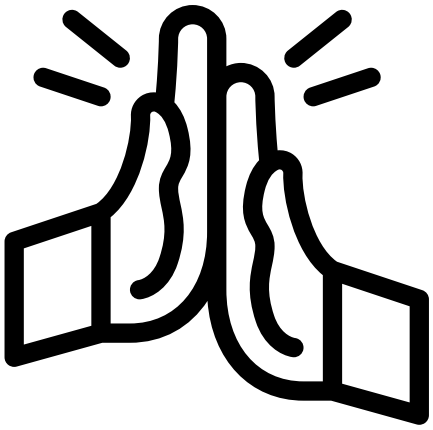
```
array([187.70399708])
```

Predicting a new result with Polynomial Regression

```
[ ]  1 lin_reg_2.predict(poly_reg.fit_transform([[2500]]))
```

```
array([187.90503589])
```

THANK YOU



You're still here and I appreciate that. Thank you for following along. I suspect you have a decent idea of how I approach retail product pricing and the nuances that are involved. Keep in mind, pricing takes many forms. There is no one-size-fits-all approach.

The key is to find a method that makes sense and is justified by data. Whatever method I decide on, it's iterative and requires maintenance and continued market intelligence to gauge where the competition is going. It's never set-it-and-forget-it. My customers help keep me in line as well. I appreciate when they are vocal about our products and pricing. It really does help me hone our prices and product benefits and features.

Thank you

Brent Janaky

janakybrent@gmail.com