**Price Optimization to Maximize Revenue**

| **Mentors*:*** | **Authors:** |
| --- | --- |
| *…..*  **Mentor (Client):**  *Sharat Chandra*  *Gomathi* | *Usharani Sahoo* |

**Table of Contents**

[***1.***](#_heading=h.3as4poj) ***Introduction 4***

[**1.1 Purpose 4**](#_heading=h.1pxezwc)

[**1.2**](#_heading=h.49x2ik5) **Scope 4**

[**1.3 Document Organization 4**](#_heading=h.2p2csry)

[**1.4**](#_heading=h.147n2zr) **Audience 4**

[***2.***](#_heading=h.3o7alnk) ***System Overview 5***

[**2.1 Context 5**](#_heading=h.23ckvvd)

[**2.2 Product Feature 5**](#_heading=h.ihv636)

[**2.3**](#_heading=h.32hioqz) **Technologies Used 5**

[***3.***](#_heading=h.1hmsyys) ***Project Pipeline 5***

[***4.***](#_heading=h.41mghml) ***Project Architecture 6***

[**4.1 Data collection 6**](#_heading=h.2grqrue)

[**4.2 Data Preparation 6**](#_heading=h.vx1227)

[**4.3**](#_heading=h.3fwokq0) **Data Dictionary & Data Understanding 6**

[**4.4**](#_heading=h.1v1yuxt) **MODEL SELECTION 7**

[***5.***](#_heading=h.4f1mdlm) ***Assumption and Constraints 8***

[***6.***](#_heading=h.2u6wntf) ***References 9***

**Revision History**

| Version | Date | Description |
| --- | --- | --- |
| 1.0 | 30/05/2022 | Initial Draft |
| *2.0* | 06/06/2022 | Final Draft |

# Introduction

## 1.1 Purpose

The purpose of this document is to outline the high-level design of the ***Price Optimization to Maximize Revenue*** and provide an overview for the tool implementation.

The project charter defines the scope, objectives, and overall approach for the work to be completed. It is a critical element for initiating, planning, executing, controlling, and assessing the project. It should be the single point of reference on the project for project goals and objectives, scope, organization, estimates, work plan, and budget. In addition, it serves as a contract between the Project Team and the Project Sponsors, stating what will be delivered according to the budget, time constraints, risks, resources, and standards agreed upon for the project.

## Scope

***Price Optimization To maximize revenue*** is an machine learning model which will predict an optimized price based on the name of the product and the zone where it will be sold. There are around 2300 products from different Brands having different material category. The sales of the product is also categorized according to 4 zones. The revenue for a product will yield maximum profit if the sales are increased. This will also help the increase the popularity of that particular product. This can be achieved by giving discounts to the customers. But giving heavy discount may affect the profits in the sales of product. Hence, an optimized selling price is to be predicted which will consider all the points discussed above.

## 1.3 Document Organization

This document is organized into the following section:

| Introduction | Provides information related to the document |
| --- | --- |
| System Overview | Describes the approach, architectural goals and constraints, Guiding principles |
| Application Architecture | Describe the application architecture in terms of different layers of application. Description of the presentation layer, business layer, data access layer and resource layer and their relationship to each other. |
| Database Architecture | Describes the overall Data model and entity relationship diagram |
| Assumptions and Constraints | Details various assumptions made during design and development of the Online Screening tool |

## Audience

The intended audiences for this document are: -

* Innodatatics Inc.
* The project development team
* Mentors

# System Overview

## 2.1 Context

The right pricing can make or break a business and copying your competitors might mean starting a price war, but making a guess could leave you balking at abysmal sales numbers.

Hence, successful price optimization is a matter of a balance that can have a major impact on your sales, customer satisfaction, profits, and achievable growth goals. “Price optimization is the process of identifying the optimal price point for any given product at any given location that will yield the highest profit”.

    Enrolled students are looking to build a machine learning model which will help the client by predicting the optimized price of a product based on the past transactions. This will help the client to maximize the sales, revenue and also keeping the profits at maximum.

## 2.2 Product Feature

The major feature of the ***Price Optimization To maximize revenue*** will be the following: -

* + **PostgreSQL** – A local server will be created to mimic the server where all the transactional data is stored. This will ensure that the model will dynamically fetch data from the server in order to get dynamic predictions.
  + **Regression Model –** The regression model will find the relation between the no of units sold and the selling price per product. This model will in turn help to calculate the revenue of that particular product.
  + **Optimization of the selling price –** A logic is developed which will calculate the revenue of a series of incremental selling prices. The revenue will be calculate based on the regression model.

## Technologies Used

The model will be developed in Python using Streamlit framework, Model will be implemented in Python.

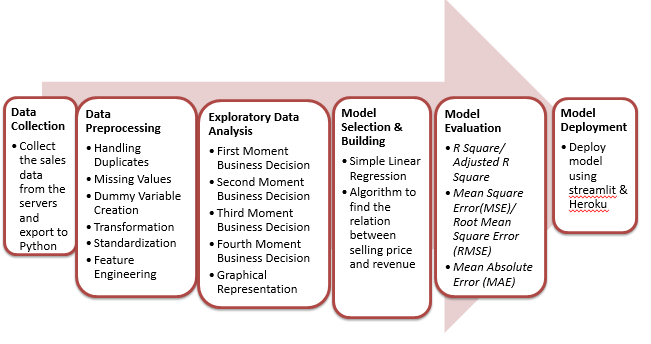
The system will have Machine Learning libraries too.

**Front-end –** HTML, CSS

**Middleware (REST API) –** Python, Streamlit framework

**Backend –** PostgreSQL

# Project Pipeline



A step by step methodology is created to ensure the smooth execution of the project. The processes are performed using PostgreSQL, Spyder and Streamlit.

# Project Architecture

## 4.1 Data collection

The transactional data is recorded from the client on a server. This server is duplicated for study purpose. Using PostgreSQL, a virtual local server is created in which a database is created. This database has all the records of transaction. The data consists of 37438 records having 4 zones, 128 material categories and 2312 different products.

## 4.2 Data Preparation

The data is checked for negative sales figures. Only non-negative records were considered for study. The data is also checked for duplicate records and NA values. All such entries were removed. The data needed to be sorted as per the different variables. Using PostgreSQL, sorted data will be fetched for model building and optimized price prediction.

## Data Dictionary & Data Understanding

Database has 37438 rows and 20 columns. The decription of the features are as follows:

| **Feature** | **Remarks** | **Dtype** |
| --- | --- | --- |
| UID | Unit ID: Id for a Material | Integer |
| NAME | Unit Name: Name of the Material | Nominal |
| ZONE | Name of the zone of Business | Nominal |
| Brand | Brand of the material | Nominal |
| MC | Material Category: Category of the material | Nominal |
| Fdate | Month of sale | Date |
| NSU | Net Sale Unit: Total units sold in a month | Float |
| NSV | Net Sale Value: Total sale value in a month | Float |
| GST Value | GST Value: GST on the NSV | Float |
| (NSV-GST) | Calculation only | Float |
| Sales at Cost | Cost to company | Float |
| SALES AT COST | Calculation only | Float |
| MARGIN% | Percentage Margin | Float |
| Gross Sales | Gross Sales | Float |
| Gross RGM(P-L) | Gross Margin | Float |
| Gross Margin % | Gross Margin Percentage = (Gross Sales – Sales at Cost – GST) / (GROSS SALE - GST) \* 100) | Float |
| MRP | Max. Retail Price of unit item | Float |
| SP | Selling price of unit item ( = NSV / NSU) | Float |
| DIS | Discount in rupees. | Float |
| DIS% | Discount Percentage (= DIS/MRP \* 100 | Float |

## MODEL SELECTION

**Simple Linear Regression:**

Regression models describe the relationship between variables by fitting a line to the observed data. Linear regression models use a straight line, while logistic and nonlinear regression models use a curved line. Regression allows you to estimate how a dependent variable changes as the independent variable(s) change.

The SLR will predict the relation between quantity and selling price. Quantity is the dependent variable where selling price SP is the independent variable.

Optimized Price:

revenue = predicted quantity \* price

where,

price is taken as a range of values between the minimum cost to company price and maximum selling price.

The following code is written to execute above logic.

def find\_optimal\_price(data\_new):

import statsmodels.api as sm

from statsmodels.formula.api import ols

# demand curve

#sns.lmplot(x = "price", y = "quantity",data = data\_new,fit\_reg = True, size = 4)

# fit OLS model

model = ols("quantity ~ price", data = data\_new).fit()

# print model summary

print(model.summary())

prams = model.params

prams.Intercept

prams.price

# plugging regression coefficients

# quantity = prams.Intercept + prams.price \* price # eq (5)

# the profit function in eq (3) becomes

# profit = (prams.Intercept + prams.price \* price) \* price - cost # eq (6)

# a range of different prices to find the optimum one

start\_price = data\_new.price.min()

end\_price = data\_new.price.max()

Price = np.arange(start\_price, end\_price,0.05)

Price = list(Price)

# assuming a fixed cost

k1 = data\_new['NSV'].div(data\_new['quantity'])

cost = k1.min()

Revenue = []

for i in Price:

quantity\_demanded = prams.Intercept + prams.price \* i

# profit function

Revenue.append((i-cost) \* quantity\_demanded)

# create data frame of price and revenue

profit = pd.DataFrame({"Price": Price, "Revenue": Revenue})

#plot revenue against price

plt.plot(profit["Price"], profit["Revenue"])

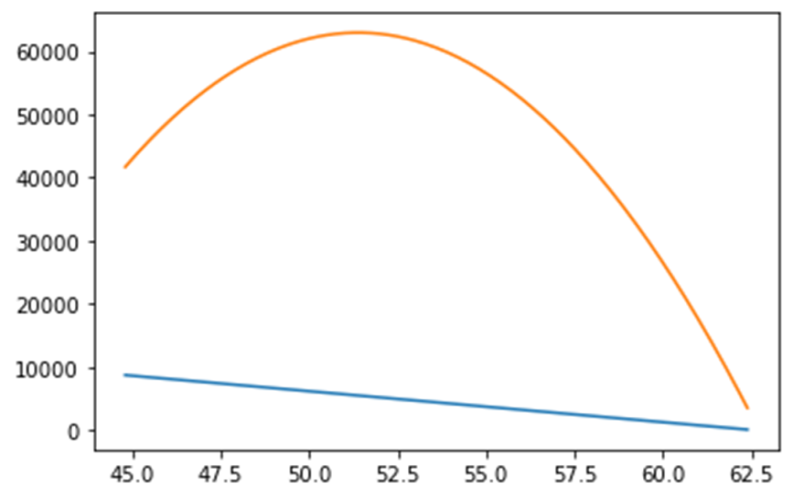
# price at which revenue is maximum

ind = np.where(profit['Revenue'] == profit['Revenue'].max())[0][0]

values\_at\_max\_profit = profit.iloc[[ind]]

return values\_at\_max\_profit

A graph is plotted between the revenue and the selling prices. A sample is shown below:



The selling price corresponding to the maximum revenue is identified as optimized selling price.

# Assumption and Constraints

* The prediction system does not cover all the products sold. The products which are sold very rarely will not have accurate predictions.
* Quick response to changing market trends.
* Price optimization for a product family -

Any changes in the pricing of one product may trigger a chain reaction across a product family. Hence, the pricing of the product family becomes a daunting task.

# References

<https://www.vendavo.com/c-suite/importance-price-optimization/>

<https://dlabs.ai/blog/price-prediction-how-machine-learning-can-help-you-grow-your-sales/>

<https://competera.net/resources/articles/machine-learning-pricing-optimization>

<https://competera.net/resources/articles/dynamic-pricing-engine>

<https://competera.net/resources/articles/dynamic-pricing-algorithm>

<https://vitalflux.com/pricing-optimization-machine-learning-techniques/>

<https://towardsdatascience.com/mercari-price-suggestion-97ff15840dbd>

<https://blog.hubspot.com/sales/price-optimization>

<https://tryolabs.com/blog/2020/06/01/price-optimization-for-e-commerce-a-case-study>