National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Sneha Muralidhar
Student ID: x22161171

School of Computing
National College of Ireland

Supervisor: Shubham Subhnil

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sneha Muralidhar……

Student ID: X22161171……

Programme: MSc Data Analytics……. **Year:** 2023-24……

Module: Research Project……

Lecturer: 14/12/2023……

Submission Due Date: ……

Project Title: Retail Price Optimization Using Machine Learning Algorithms……

Word Count: 500…… **Page Count:** 5……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sneha Muralidhar……

Date: 14/12/2023……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sneha Muralidhar
Student ID: x22161171

Welcome to the Configuration Manual for Retail Price Optimization! Your go-to resource for configuring and using the Jupyter Notebook "Price Optimization.ipynb" is this guide. This user-friendly manual offers detailed instructions on how to set up your system, upload the "retail_price" dataset, and use machine learning to optimize retail prices.

Whether you work in retail, data science, or development, these clear instructions will help you maximize the potential of the Decision Tree Regressor, modify pricing plans based on market conditions, and make wise choices. Let's maximize retail prices to increase sales and satisfy customers. Now let's get going!

1 System Requirements

A. Hardware Requirements

Processor: i5 processor
RAM: 16GB
Storage: 256GB
Operating System: Windows

B. Software Requirements

Google Colab is used to execute the code. Colab is a web hosted Jupyter notebook service that offers free access to computing resources, such as GPUs, without the need for setup. Similarly Jupyter Notebook can also be used.

Python version: 3.6

C. Libraries and Packages

Numerous widely used Python libraries are preinstalled on Google Colab. The following Python packages are installed using pip and used to implement the project:

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Plotly
- Sklearn

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
import os
import math
import warnings

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score
```

2 Setting up Google Collab

1. Launch the Google Colab browser window.
2. Either import the pre-existing "Retail Price Optimization.ipynb" notebook or create a new one.
3. Run the following code cell to make sure your Google Drive is mounted to the Colab notebook:

```
from google.colab import drive
drive.mount('/content/drive')
```

3 Uploading Dataset

1. Run the following code to upload the dataset from device.

```
from google.colab import files
x = files.upload()
```

2. Upload the dataset “reatil_price” and run the code.

4 Data Preprocessing

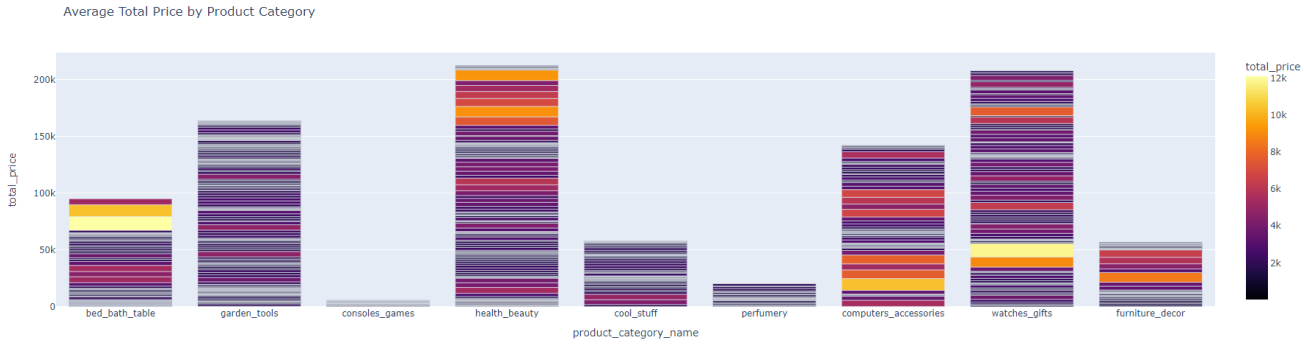
Ensure that the dataset has been preprocessed in accordance with the notebook's instructions, which cover handling missing values, identifying outliers, normalizing, and encoding categorical variables.

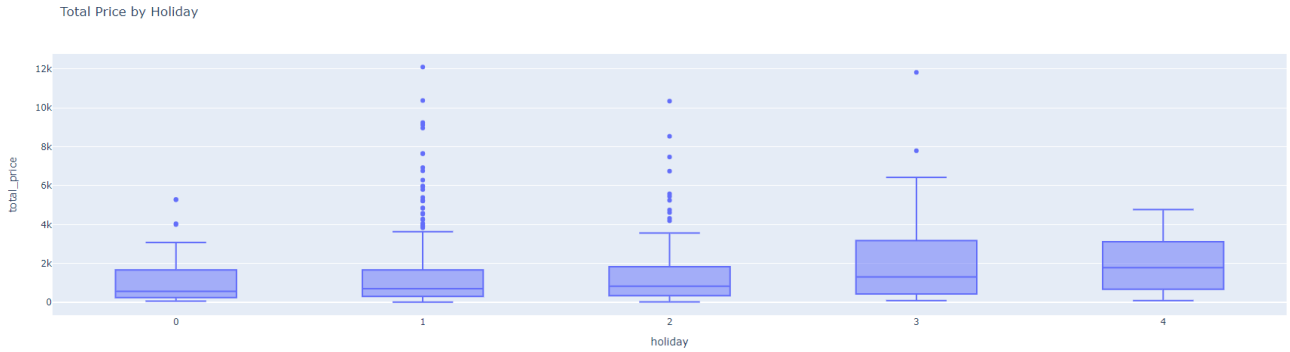
```
print(dataset.isnull().sum())
```

```
product_id                    0
product_category_name         0
month_year                    0
qty                           0
total_price                   0
freight_price                 0
unit_price                    0
product_name_lenght           0
product_description_lenght    0
product_photos_qty            0
product_weight_g              0
product_score                 0
customers                     0
weekday                       0
weekend                       0
holiday                       0
month                         0
year                          0
s                             0
volume                        0
comp_1                        0
ps1                           0
fp1                           0
comp_2                        0
ps2                           0
fp2                           0
comp_3                        0
ps3                           0
fp3                           0
lag_price                     0
dtype: int64
```

5 Explanatory Data Analysis

EDA is conducted to understand and visualize the data. Some of the examples are shown below.





6 Model Training

Dataset is split into 80% of training data and 20% of test data to train the Decision Tree Regressor model.

```
X = dataset[['qty', 'unit_price', 'comp_1',
          'product_score', 'comp_price_diff']]
y = dataset['total_price']

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)

# Training a linear regression model

model = DecisionTreeRegressor()
model.fit(X_train, y_train)
```

7 Hyperparameter tuning.

Two evaluations of the model are conducted. Twice, using the factory settings and a second time with the hyperparameters adjusted.

Model 2

```
X = dataset[['qty', 'unit_price', 'comp_1',
          'product_score', 'comp_price_diff']]
y = dataset['total_price']

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)

# Training a linear regression model

model_two = DecisionTreeRegressor(
    criterion='squared_error',  # mean squared error
    splitter='best',   # choose the best split
    max_depth=10,    # unlimited depth
    min_samples_split=5,
    min_samples_leaf=2,
    max_features=None,  # consider all features
    random_state=42     # for reproducibility
)
model_two.fit(X_train, y_train)
```

8 Model Evaluation

Evaluation metrics such as R2_Score, Mean Absolute Error are used to evaluate the performance of the model.

Model Evaluation

```
r2 = r2_score(y_test, y_pred)
print(f"R-squared (R2) score: {r2}")

# Calculate the Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f"Mean Absolute Error (MAE): {mae}")
```

```
R-squared (R2) score: 0.9532337391390635
Mean Absolute Error (MAE): 152.7873529411765
```