

Prediction Product AD Campaign Performance

Project: 5



June 15, 2024

spinnaker analytics

**Project Report: Prediction Product AD Campaign Performance**

**Abstract**

This project explores the effectiveness of various marketing campaigns through extensive data analysis. By leveraging data-driven insights, the aim is to optimize marketing strategies, allocate budgets effectively, and enhance campaign performance.

**Table of Contents**

1. **Introduction**
2. **Importing Libraries & Data**
3. **Basic Analysis**
4. **Data Preprocessing**
5. **Exploratory Data Analysis**
6. **Regression Analysis**
7. **Model Building**
8. **Conclusion**

**1. Introduction**

The primary objective of this project is to analyze the performance of marketing campaigns aimed at predicting product AD performance. The dataset includes comprehensive information on campaign types, target audiences, channels used, conversion rates, ROI, and customer segments.

**2. Importing Libraries & Data**

Libraries such as pandas, numpy, matplotlib, and seaborn were utilized for data manipulation, analysis, and visualization. The dataset, sourced from products\_campaign\_sales.csv, comprises 200,000 rows and 16 columns.

**3. Basic Analysis**

Initial exploration of the dataset revealed:

* No missing values
* 16 columns including categorical features, necessitating encoding into numeric forms for analysis.

**4. Data Preprocessing**

Categorical columns (limit\_infor, campaign\_type, campaign\_level, product\_level) were converted to categorical data types for optimized memory usage and efficiency.

**5. Exploratory Data Analysis**

**Key Visualizations:**

* **Distribution of Campaign Types**: Analyzed using pie charts to depict the proportion of various campaign types.
* **Histograms and Box Plots**: Visualized orders, price distribution, and price variation across different campaign types.
* **Scatter and Line Plots**: Explored relationships such as price vs. orders and price over time (simulated by hour\_resouces) with campaign type as a variable.
* **Bar Plots**: Highlighted average orders by campaign type and product level distribution across campaign types.
* **Correlation Matrix**: Computed correlations among numeric variables (limit\_infor, campaign\_type, etc.) to identify relationships.

**6. Regression Analysis**

Implemented Linear Regression to predict Conversion\_Rate based on Duration, Clicks, Impressions, and Acquisition\_Cost. Evaluation metrics such as Mean Squared Error and R-squared score were used to assess model performance.

**7. Model Building**

Utilized a pipeline with preprocessing steps including One-Hot Encoding for categorical features (Campaign\_Type, Target\_Audience, etc.) and a Linear Regression model to predict Conversion\_Rate.

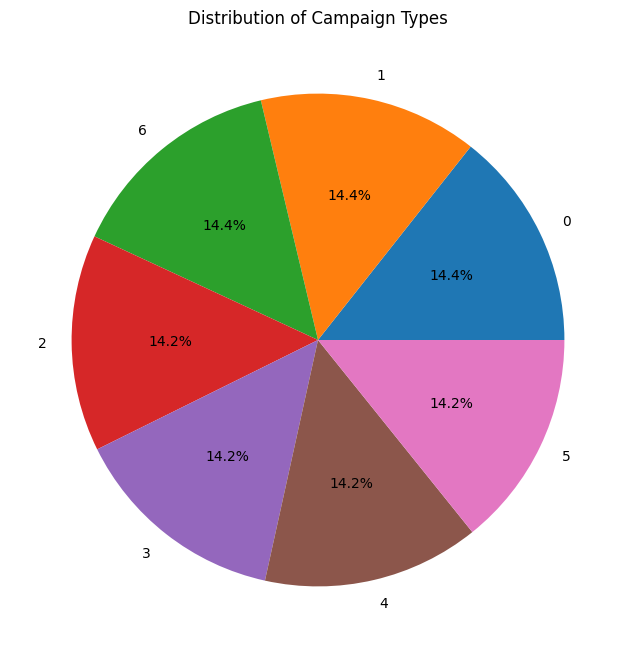
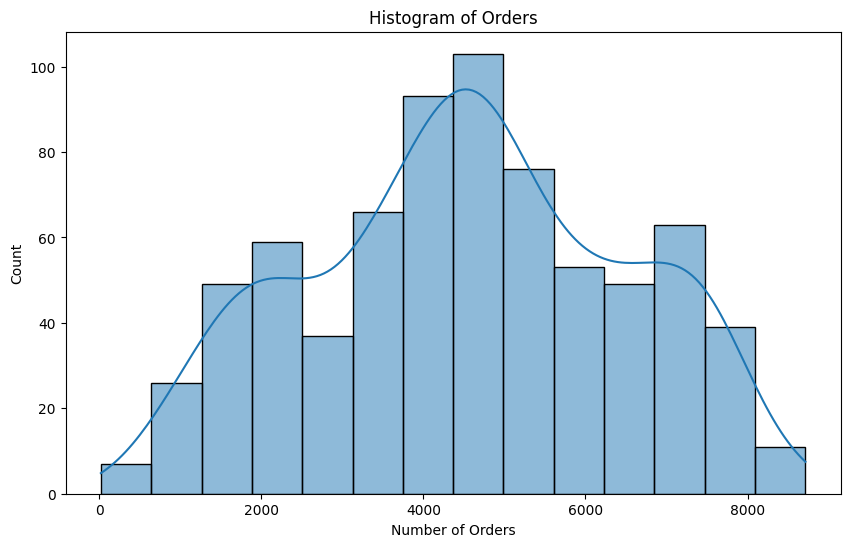
**8. Conclusion**

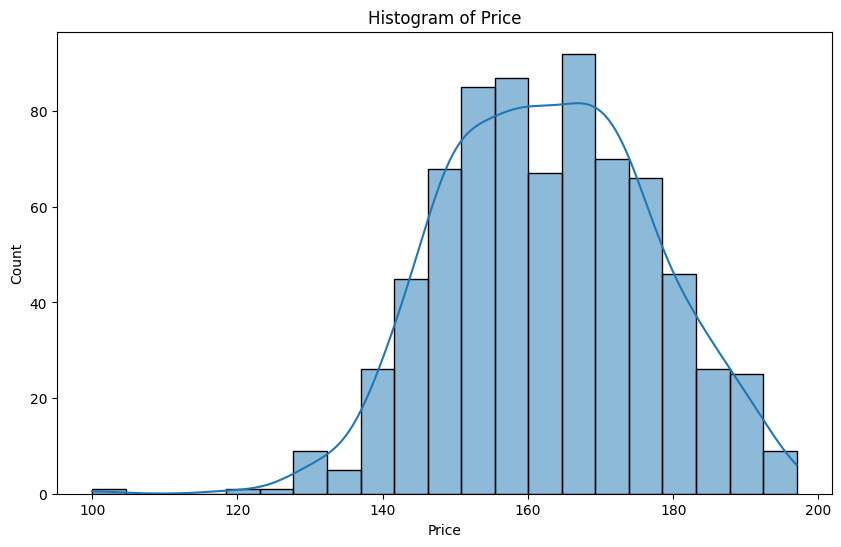
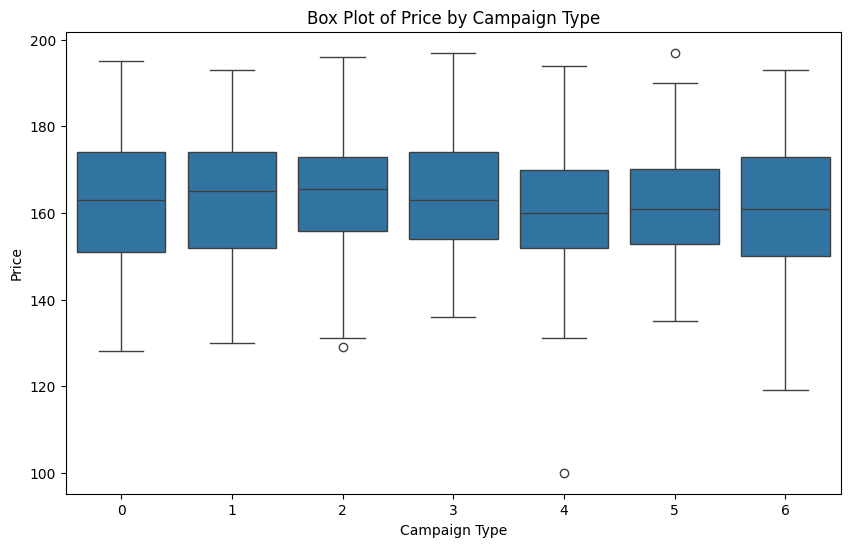
Through comprehensive data analysis and visualization, this project identified several key insights:

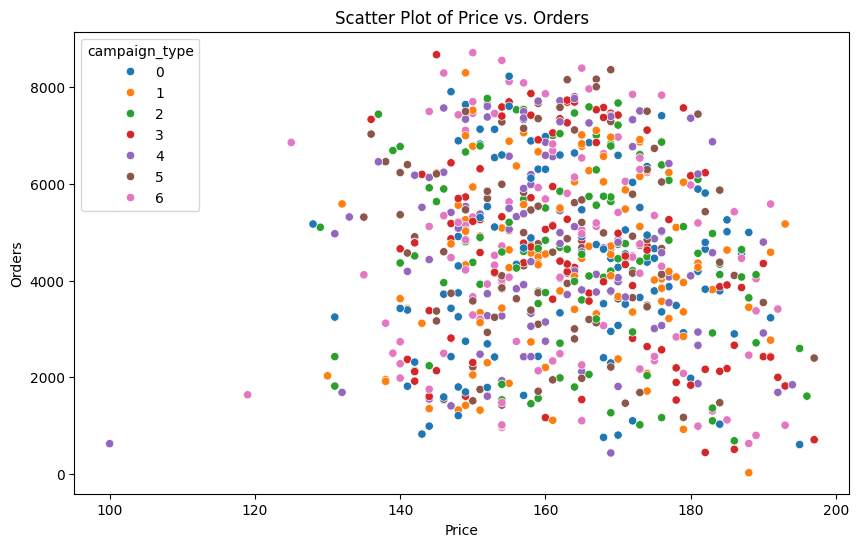
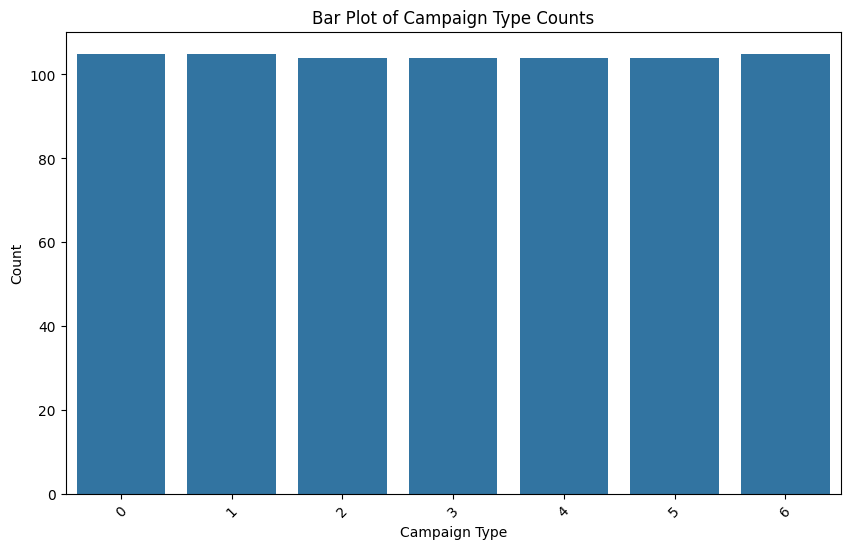
* **Campaign Types**: Evaluated performance across different types, suggesting avenues for further optimization.
* **Target Audience and Channels**: Explored preferences and effective channels for engagement.
* **ROI and Cost Efficiency**: Analyzed profitability and efficiency metrics to guide budget allocation.
* **Customer Segments**: Segmentation insights facilitated tailored marketing strategies.

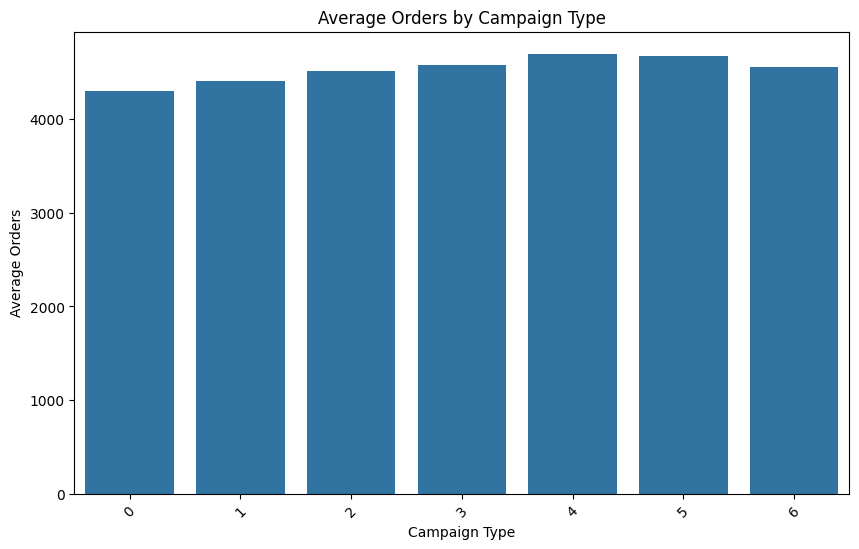
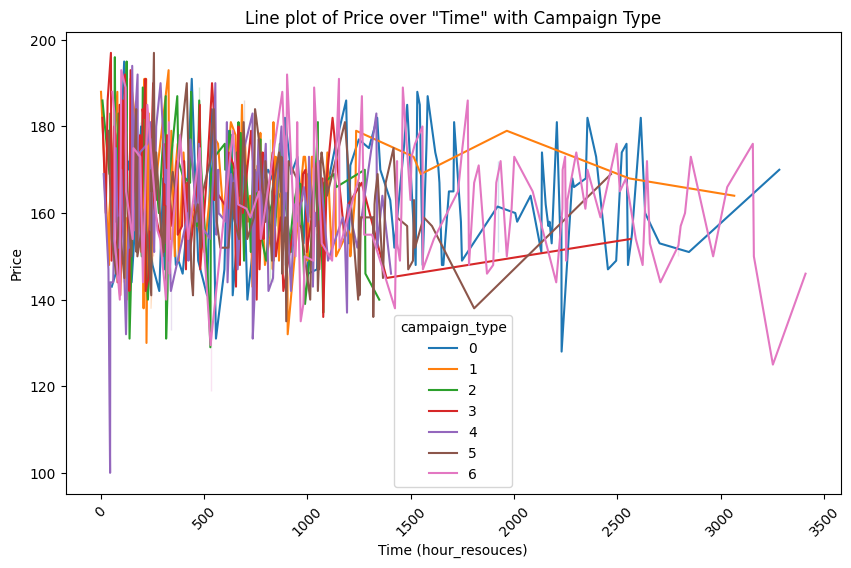
These findings enable informed decision-making to enhance marketing campaign effectiveness, optimize resource allocation, and achieve targeted growth.

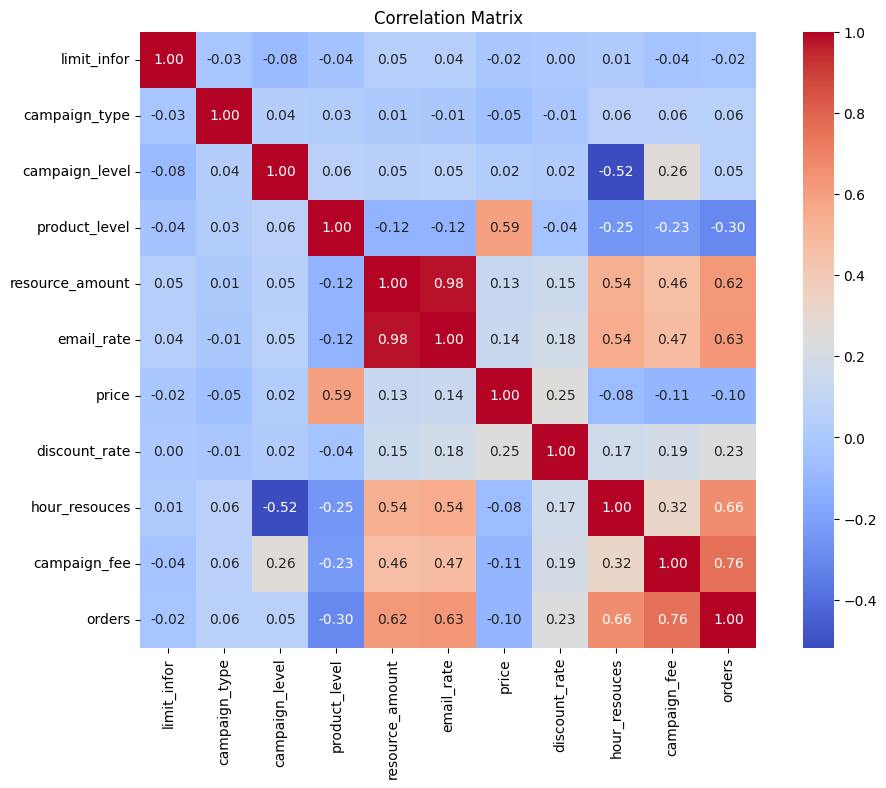
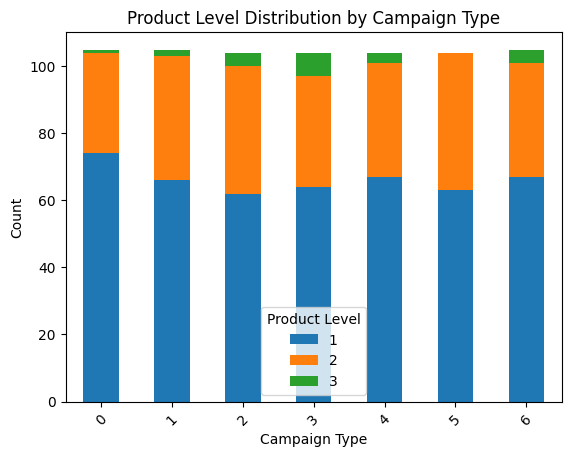
* 1. **Charts and Graphs**

** **





**11. Source codes**

<h1><center style="color:#FFFF00; font-family:bold;">Prediction Product AD Campaign Performance</center></h1>

\*\*\*

# Introduction

This notebook explores the \*\*Prediction Product AD Campaign Performance\*\*, aiming to uncover valuable insights into the effectiveness of various marketing campaigns. The dataset contains extensive information about campaign types, target audiences, channels used, conversion rates, acquisition costs, ROI, locations, languages, clicks, impressions, engagement scores, customer segments, and dates.

Understanding and analyzing these metrics can provide marketers and data analysts with crucial information to optimize marketing strategies, allocate budgets effectively, and enhance campaign performance. By leveraging data-driven insights, businesses can tailor their approaches to engage their target audiences more effectively and achieve higher returns on their marketing investments.

In this exploration, we will delve into exploratory data analysis (EDA), visualize key metrics, conduct correlation analyses, and build predictive models to extract actionable insights. By the end, we aim to empower stakeholders with the knowledge needed to make informed decisions that drive growth and success in their marketing campaigns.

# 📚 Importing Libraries & Data

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

df= pd.read\_csv("products\_campaign\_sales.csv")

print("Number of datapoints:", len(df))

# 🔢 Basic Analysis

df.head()

df.tail()

From the given dataframe above, looks like the data doesn't need further cleaning.

df.info()

From the data above, we can observe the following:

- There are no missing values

- There are 16 columns and 200000 rows

- There are some categorical features in our data frame; as there are some features in dtype: object). So we will need to encode them into numeric forms later.

# 🛠️ Data Preprocessing

df.head()

# Check for any missing values

df.isnull().sum()

# Summary statistics

df.describe()

# Convert specific columns to categorical dtype

to\_convert = ['limit\_infor', 'campaign\_type', 'campaign\_level', 'product\_level']

for col in to\_convert:

df[col] = df[col].astype('category')

# Display the updated data types to confirm conversion

df.info()

# Confirm the first few rows of the processed dataset

df.head()

# Display the number of unique values for each column

unique\_values = df.nunique()

print("Number of unique values for each column:\n", unique\_values)

# Display the unique values for each column

for column in df.columns:

unique\_vals = df[column].unique()

print(f"\nUnique values in '{column}':")

print(unique\_vals)

df.info()

# 🔎 Exploratory Data Analysis 📊

Let's perform some exploratory data analysis (EDA) and create visualizations for the given dataset.

df.head()

# Count the occurrences of each campaign type

campaign\_type\_counts = df['campaign\_type'].value\_counts()

# Create a pie chart to visualize the distribution of campaign types

plt.figure(figsize=(8, 8))

plt.pie(campaign\_type\_counts, labels=campaign\_type\_counts.index, autopct='%1.1f%%')

plt.title('Distribution of Campaign Types')

plt.show()

# Histogram of Orders

plt.figure(figsize=(10, 6))

sns.histplot(data=df, x='orders', kde=True)

plt.title('Histogram of Orders')

plt.xlabel('Number of Orders')

plt.ylabel('Count')

plt.show()

# Histogram of Price

plt.figure(figsize=(10, 6))

sns.histplot(data=df, x='price', kde=True)

plt.title('Histogram of Price')

plt.xlabel('Price')

plt.ylabel('Count')

plt.show()

# Box plot of Price by Campaign Type

plt.figure(figsize=(10, 6))

sns.boxplot(data=df, x='campaign\_type', y='price')

plt.title('Box Plot of Price by Campaign Type')

plt.xlabel('Campaign Type')

plt.ylabel('Price')

plt.show()

# Scatter plot of Price vs. Orders

plt.figure(figsize=(10, 6))

sns.scatterplot(data=df, x='price', y='orders', hue='campaign\_type')

plt.title('Scatter Plot of Price vs. Orders')

plt.xlabel('Price')

plt.ylabel('Orders')

plt.show()

# Bar plot of Campaign Type counts

plt.figure(figsize=(10, 6))

sns.countplot(data=df, x='campaign\_type')

plt.title('Bar Plot of Campaign Type Counts')

plt.xlabel('Campaign Type')

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.show()

Line plot of Price over "Time" with Campaign Type

# Sort the dataframe by a relevant column (e.g., 'hour\_resouces') to simulate 'Date'

df\_sorted = df.sort\_values('hour\_resouces')

# Line plot of Price over 'Date' (simulated by 'hour\_resouces'), with 'campaign\_type' as hue

plt.figure(figsize=(10, 6))

sns.lineplot(data=df\_sorted, x='hour\_resouces', y='price', hue='campaign\_type')

plt.title('Line plot of Price over "Time" with Campaign Type')

plt.xlabel('Time (hour\_resouces)')

plt.ylabel('Price')

plt.xticks(rotation=45)

plt.show()

\*\*Average Orders by Campaign Type\*\*

# Group by 'campaign\_type' and calculate the average 'orders'

campaign\_orders = df.groupby('campaign\_type')['orders'].mean().reset\_index()

# Create a bar plot using the average 'orders'

plt.figure(figsize=(10, 6))

sns.barplot(data=campaign\_orders, x='campaign\_type', y='orders')

plt.title('Average Orders by Campaign Type')

plt.xlabel('Campaign Type')

plt.ylabel('Average Orders')

plt.xticks(rotation=45)

plt.show()

\*\*Product Level Distribution by Campaign Type\*\*

# Replace 'Campaign\_Type' and 'Customer\_Segment' with your actual column names from the dataset

segment\_campaign = pd.crosstab(df['campaign\_type'], df['product\_level'])

# Create a stacked bar chart

plt.figure(figsize=(10, 6))

segment\_campaign.plot(kind='bar', stacked=True)

plt.title('Product Level Distribution by Campaign Type')

plt.xlabel('Campaign Type')

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.legend(title='Product Level')

plt.show()

# Correlation matrix

# Assuming you want to include specific numeric columns for correlation analysis

numeric\_cols = ['limit\_infor', 'campaign\_type', 'campaign\_level', 'product\_level', 'resource\_amount',

'email\_rate', 'price', 'discount\_rate', 'hour\_resouces', 'campaign\_fee', 'orders']

# Selecting only the numeric columns

numeric\_df = df[numeric\_cols]

# Compute the correlation matrix

corr\_matrix = numeric\_df.corr()

# Plotting the correlation matrix using Seaborn

plt.figure(figsize=(12, 8))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt='.2f', square=True)

plt.title('Correlation Matrix')

plt.show()

# Regression Analysis

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.impute import SimpleImputer

# Load your dataset

df = pd.read\_csv("marketing\_campaign\_dataset.csv")

# Remove dollar sign ('$') from 'Acquisition\_Cost' if necessary

df['Acquisition\_Cost'] = df['Acquisition\_Cost'].replace('[\$,]', '', regex=True)

# Convert 'Acquisition\_Cost' to numeric if necessary

df['Acquisition\_Cost'] = pd.to\_numeric(df['Acquisition\_Cost'], errors='coerce')

# Identify and handle non-numeric values in 'Duration' column

# For example, replace '30 days' with NaN

df['Duration'] = pd.to\_numeric(df['Duration'], errors='coerce')

# Assuming these are the correct columns in your dataset

X\_columns = ['Duration', 'Clicks', 'Impressions', 'Acquisition\_Cost']

y\_column = 'Conversion\_Rate' # Assuming 'Conversion\_Rate' is the target variable you want to predict

# Verify if the columns exist in your dataset

for col in X\_columns + [y\_column]:

if col not in df.columns:

raise KeyError(f"Column '{col}' not found in the dataset.")

# Prepare X and y

X = df[X\_columns]

y = df[y\_column]

# Handle missing values with SimpleImputer

imputer = SimpleImputer(strategy='median') # Use median strategy for missing values

X\_imputed = imputer.fit\_transform(X)

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_imputed, y, test\_size=0.3, random\_state=42)

# Train the model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'Mean Squared Error: {mse}')

print(f'R^2 Score: {r2}')

# Model Building

# Define categorical features

categorical\_features = ['Campaign\_Type', 'Target\_Audience', 'Channel\_Used', 'Location', 'Language']

# Pipeline for preprocessing categorical features

preprocessor = ColumnTransformer(

transformers=[

('cat', OneHotEncoder(), categorical\_features)

])

# Prepare X and y

X = df.drop(['Conversion\_Rate'], axis=1) # Features

y = df['Conversion\_Rate'] # Target variable

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Initialize the linear regression model

model = LinearRegression()

# Create a pipeline with preprocessing and modeling steps

pipeline = Pipeline(steps=[

('preprocessor', preprocessor),

('model', model)

])

# Fit the pipeline on the training data

pipeline.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = pipeline.predict(X\_test)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'Mean Squared Error: {mse}')

print(f'R^2 Score: {r2}')

# Conclusion💥

In this analysis, we explored the Marketing Campaign Performance Dataset and gained valuable insights into the effectiveness of various marketing campaigns. Here are the key findings from the analysis:

\* \*\*Campaign Types:\*\* The dataset includes various campaign types such as email, social media, influencer, display, and search. Further analysis can be conducted to evaluate the performance of each campaign type and identify the most effective ones for different customer segments.

\* \*\*Target Audience:\*\* The dataset provides information about the specific audience segments targeted by the campaigns. Understanding the preferences and characteristics of different target audiences can help in tailoring marketing strategies to effectively engage and convert potential customers.

\* \*\*Channel Usage:\*\* The dataset includes information about the channels used to promote the campaigns, such as email, social media platforms, YouTube, websites, and Google Ads. Analyzing channel effectiveness can help in optimizing marketing efforts by focusing on the channels that generate higher conversion rates and engagement.

\* \*\*ROI and Acquisition Cost:\*\* The ROI (Return on Investment) and acquisition cost metrics provide insights into the profitability and cost-efficiency of the campaigns. By analyzing these metrics, marketers can identify the campaigns with the highest ROI and optimize their marketing budget allocation.

\* \*\*Customer Segments:\*\* The dataset categorizes campaigns based on specific customer segments such as tech enthusiasts, fashionistas, health and wellness enthusiasts, foodies, and outdoor adventurers. Understanding the preferences and behavior of different customer segments can aid in creating personalized and targeted marketing campaigns.

By leveraging the insights gained from this analysis, marketers and data analysts can refine their marketing strategies, optimize campaign performance, and drive targeted growth. The findings from this analysis can guide data-driven decision-making and support market research efforts.