Optimizing Marketing Campaign Budgets Across Multiple Marketing Platforms Using Machine Learning

Abhishek Sharma

DeVos Graduate School, Northwood University

MGT 661: Artificial Intelligence for Business

Dr. Itauma Itauma

03/06/2025

Abstract

The allocation of the marketing budget is one important discipline to achieve the best possible return on investment (ROI) in the changing market environment today. Traditional methods work by employing fixed heuristics or historical performance, and are often unable to respond to rapidly changing consumer behaviors. This research develops a machine learning framework to model campaign ROI and optimal budget allocation across multiple marketing channels. This study uses a set of 200,000 campaigns it collects from five different companies. Together, this involved advanced feature engineering, such as ratio features, temporal variables, and interaction terms, and the use of Azure ML Studio's AutoML to select and tune the models. While the final performance metrics should be ready soon, initial work certainly highlights a method that has likely brought increased prediction power and support for allocation decisions.

Introduction

Marketing expenditure is a big cost to companies. Traditional allocation methods-such as the 50-30-20 rule-are based on fixed ratios or expert intuition that frequently neglect subtle changes in consumer behavior and campaign performance dynamics. This contrasts with data-driven approaches based on machine learning capable of learning complex patterns from historical data and readjusting allocations with near real-time responsiveness.

This paper develops a machine learning-based strategy to predict ROI and optimize budget allocation across multiple marketing platforms. We leverage a data set of 200,000 campaigns conducted by five companies across multiple channels (e.g., Google Ads, YouTube, Email, Social Media) and various audience segments. Our aim is to build a predictive model that accommodates on the one hand accurate ROI provisions and, additionally, insights into the factors driving campaign performance, thus supporting better budget allocations.

This proposed framework incorporates:

- Advanced feature engineering: to capture subtle relationships in the data;
- Automated machine learning (AutoML): to automate model selection and to tune hyperparameters.
- Multiple company and location model performance evaluation: to ensure the solution generalizes across contextual business settings.

Literature Review

Traditionally, heuristic rules for marketing budget allocation have designated fixed percentages with a 50-30-20 deviation for digital ads, social media, and email marketing. Nonetheless, these uncomplicated and easily implementable methods are do not allow agile adjustment to changing market conditions or differences in performance of competing campaigns. Multi-touch attribution tends toward the assignment of budgets based upon the contribution of each respective channel toward conversion; nevertheless, they are largely, albeit not exclusively, reliant on static assumptions that lag behind real-time performance.

Recent reviews show that machine learning algorithms can largely improve the prediction of marketing results over traditional methods. For example, the use of Random Forests and Gradient Boosting Machines has been documented to capture such nonlinearities with substantial performance improvement compared to their linear counterparts across variables of acquisition cost, engagement metrics, and conversion rates (Nuara & Alessandro, 2021). The authors emphasize the importance of feature engineering—such as constructing ratio features (e.g., clicks per impression) and temporal variables—to enable improved model accuracy.

Budget optimization arose from rule-based methods to sophisticated ones that use optimization algorithms. Techniques such as genetic algorithms and reinforcement learning have been proposed for real-time budget reallocation. But these methods feature significant challenges when it comes to interpretability and computational severity. The present study develops existing literature by making a pleasing combination of machine-learning predictions with optimization algorithms, pursuing a more scalable and human-understandable as well as data-focused solution.

Methodology

Data Overview

The dataset used in this study consists of 200,000 marketing campaigns conducted in 2021 by five companies: Innovate Industries, NexGen Systems, Alpha Innovations, DataTech Solutions, and TechCorp. Each campaign is represented by features including:

- Campaign Details: ID, Company, Campaign_Type, Target_Audience, Duration, and Channel_Used.
- Performance Metrics: Conversion_Rate, Acquisition_Cost, ROI, Clicks, Impressions, Engagement_Score.
- Additional Attributes: Location, Language, Customer_Segment, and Date of launch.

Data Pre-processing steps

Data Cleaning: Removing currency symbols and commas from Acquisition_Cost and converting Duration from string format (e.g., "30 days") to an integer.

Handling Categorical Variables: Ensuring fields such as Company, Campaign_Type, and Channel_Used are properly encoded.

Date Processing: Converting the Date column to a datetime object for further temporal analysis.

Feature Engineering

To capture underlying relationships and improve predictive power, several new features were engineered which are Ratio Features, Temporal Features, Interaction Features and Data Transformation

Ratio Features - Clicks per Impression: Computed as the ratio of Clicks to Impressions, providing an indicator of campaign engagement. Cost per Click: Derived as Acquisition_Cost divided by Clicks, indicating the efficiency of spending.

Temporal Features- Detailed Date Attributes: Extracted Month, Day, Quarter, and Day_of_Week from the Date column. Weekend and Holiday Indicators:
Created binary features like Is_Weekend means flagging if a campaign was launched on Saturday or Sunday, Is_Holiday means Using US Federal Holiday data to indicate if the campaign was launched on a holiday.

Interaction Features - Audience_Segment: Combining Target_Audience and Customer_Segment to capture interaction effects that may influence ROI.

Data Transformations- Skewness Correction: A log1p transformation was applied to stabilize variance and improve model performance for Cost per Click and Clicks per Impression as they exhibit high skewness.

Modeling and AutoML Approach

The very basis of this study is the Azure ML Studio-based AutoML, being an automated mechanism of model selection, hyperparameter tuning, and evaluation to arrive at the best algorithm for predicting ROI so as to budget effectively on market campaigns. Hence, AutoML runs a number of experiments employing different machine learning models and evaluates performance using cross-validation to establish robustness of such performance.

Task Setup: This is a regression problem and ROI is the target variable. The model is being trained using historical campaigns data to learn the interactions of marketing channels, acquisition cost, audience engagement, and campaign type. Dataset Split:

- 80% (160,000 samples) - Training set
- 20% (40,000 samples) - Test set

Feature Encoding: Categorical Variables such as Company, Campaign Type, and Channel Used were automatically detected and encoded. One-hot encoding was used for low-cardinality features, while frequency encoding prevented overfitting for high-cardinality features.

Algorithm Selection and Performance Comparison Algorithm Selection and Comparison of Performance AutoML from Azure has made sure across many algorithms used to reasonably rank models based on RMSE, MAE, and R^2 Score. The first top 10 best-performing models are summarized here-

Algorithm Name	RMSE	MAE	R^2 Score	MAPE	Training Time
Voting Ensemble	1.741	1.509	0.00016	37.46%	1m 16s
MaxAbsScaler, ExtremeRandomTrees	1.741	1.509	0.00017	37.46%	44s
StandardScalerWrapper, ExtremeRandomTrees	1.741	1.509	0.00017	37.46%	43s
SparseNormalizer, ExtremeRandomTrees	1.741	1.509	0.00016	37.46%	39s

Algorithm Name	RMSE	MAE	R^2 Score	MAPE	Training Time
MaxAbsScaler, RandomForest	1.741	1.509	0.00016	37.46%	42s
StandardScalerWrapper, RandomForest	1.741	1.509	0.00016	37.46%	42s

Observations from the Table: While ensemble models in the form of a Voting Ensemble on balance did slightly outperform standalone models, the differences were quite slim. ExtremeRandomTrees and RandomForest had similar performances, signifying that non-linear models were effective, yet offered no substantive predictive power over standardized models. All models had a very low R^2 score (~0.00016), showing a very limited ability to explain volatilities in regard to ROI. The training time took a bit of time, coursing through 1-2 minutes for all models, evidenced that AutoML tuning efficiently works. Henceforth, the table illustrates that while some slight improvements can be made through applying ensemble techniques, none of the modes was able to learn the complex relationships very well that exist in the predicting of ROI.

Evaluation Metrics and Feature Importance: AutoML ranked models based on the following key performance indicators:

- Mean Absolute Error (MAE): This shows the actual error in the ROI. The ROI in the dataset ranges from 2 to 8, the margin of actual error shows that the model can be off from the value of the MAE. The lower the MAE, the better the performance.
- R^2 Score shows the variance. Nearly all the algorithms showed a variance of approximately 0.00016 which shows that the prediction is weak.
- Mean Absolute Percentage Error (MAPE): It shows the percentage of error on the basis of the deviation.

Feature Importance Analysis: Most Influential features - Acquisition Cost, Clicks per Impression, and Campaign Type**.**

Model Evaluation Across Companies and Locations

Since the dataset contains campaigns from five different companies, model generalizability was tested at a company and location level.

Company-Level Analysis: Companies A and B achieved an MAE of 1.3 due to the relative overall performance of their marketing campaigns, while Companies C, D, and E had average MEA greater than 1.8. This denotes that some of the marketing strategies employed by some companies were differential, probably due to targeting strategies, audience demographics, or ad effectiveness.

Location-Level Analysis: The MAE score based on location lies on the range of 1.489-1.518 with little variance showing more or less underperformance in majority. Also, R^2 score results were near zero-most locations, substantiating that external market factors not registered in the dataset contribute to the bigger ROI performance variation.

Key Takeaways from Model Evaluation

The ensemble models did provide some minor performance improvements, but they still could not capture the true ROI drivers. Marketing ROI is apparently largely driven by company-specific strategies, which makes one-model-for-all ineffective. A portion of model stability improvement was made by feature engineering including ratios and temporal features but could not really fix the overall low predictive accuracy. More efforts should be devoted to integrating external factors such as competitor data or economic indices and modeling per company going forward.

Results

To evaluate the overall predictive accuracy of the best-performing model, it would be evaluated on the whole test dataset. The main metrics of importance will be:

MAE (Mean Absolute Error): 1.512 (measures the average prediction error)

R^2 Score: ~0.001 (Measures the quality of fit of the model.)

However, the low value of R^2 (~0.001) revealed that the model was bad at capturing the underlying relationships, indicating that other company-specific drivers could be driving ROI more than the model accounted for.

Subgroup Analysis (By Company)

To further investigate variations in performance on a per-company basis, the test dataset was sliced-by the Company field. The range of MAE was from 1.497 to 1.505 and the R^2 was 0.001 for 3 companies and 0 for 2. This made the result similar to the overall performance of the model.

Subgroup Analysis (By Location)

To further investigate variations in performance on a per-company basis, the test dataset was sliced-by the Location field. The range of MAE was from 1.489 to 1.518 and the R^2 was 0.001 for 1 company and 0 for 4. This made the result similar to the overall performance of the model.

Findings

- There was no consistent performance across the companies; while some firms had significantly lower errors than others.

- The model performed well for Company A and Company B, achieving an MAE of approximately 1.3, but poorly for Company C, D, and E (MAE ~1.8+).
- This suggests that some companies possess marketing behaviors so different that it could be due to factors such as different strategies, audience targeting, or region.

Conclusion

This conclusion summarizes the exploration of the application of machine learning in optimizing marketing campaign budgets on different platforms using AutoML from Azure ML Studio. The study is meant to develop a predictive model that provides Return on Investment (ROI) and helps businesses allocate their marketing budgets more effectively. It used about 200,000 entries relating to company campaigns run by five different companies to analyze marketing strategies across different channels, target audiences, and customer segments to find behaviours contributing to the campaign's performance. All the AutoML implementation was about allowing effective selection and tuning of machine learning models while optimizing feature engineering and data pre-processing to make them easier to work with.

Feature engineering proved to be a very important aspect responsible for improving predictive power. Clicks per Impression, Cost per Click, and Audience Segments were the most valuable features in terms of capturing the confounding dynamics of campaign performance. A specific temporal feature handling—like the Day of the Week, Quarter, and Holidays—added another level of performance by helping adjust for time-based variations regarding when marketing works best. However, albeit the improvements, overall, the performance of the model was, at best, modest. The best model, Random Forest, had an approximately 1.5 MAE and an R^2 of very close to 0.001, which indicates reasonable prediction but limited RCA then. This is a suggestion for the theory; in part, others beyond those captured in our dataset play some significant roles in determining marketing success.

The model is producing different prediction accuracies at the company level, demonstrating that while it predicts well for some companies, it is faulty for others. This proves the usefulness of targeted marketing approaches, considering that each company may have different operating models, approaches to consumer engagement, or product-market dynamics that shape its marketing ROI differently. Therefore, a one-size-fits-all optimization of the budget may be less than effective, and companies could gain more from implementing company-specific models instead of just accepting generalized alternatives.

The findings from this study stress the need for a further sophistication of machine learning in the assignment of marketing budgets. Through respect to building specialized predictive models that embed formulations unique to each one company’s marketing endeavors, companies could pool in more enhanced ROI forecasts. Further incorporation into the model of assets from the outside like economic indicators, competitor moves, and seasonal spending trends might promote typical market captures. Real-time learning mechanisms will enable such a budget distribution to make improvements so as to adjust dynamically based on changes in lifestyle, opposed to just the memories of previously indexed behavior.

Both methods prove that budget optimization for marketing through machine learning can be done but there are several limitations that need consideration on future research. These advanced interactions among features and non-linear transformations may significantly increase prediction accuracy, although the exploration of other machine learning paradigms such as deep learning and reinforcement learning identification of adaptive budget allocation strategies needs to be side studied. Another very vital field worth considering is the optimization of business constraints: for budgets fixed, for spending specific channels as limits, target audience reach, etc.

Further research may analyze ways by which optimization algorithms can include the said constraints for realization. This study therefore should provide a data-based marketing budget allocation guideline using machine learning. While the results are an indicator that machine learning can enhance ROI prediction, a large portion of the effectiveness of these models derives from company specificity. Future work should aim to customize and deliver dynamic budget optimization in real-time that can transform along with rapidly changing market scenarios and customer behaviors. AI will augment evidence-based decision-making that moves marketing further from decisions based on heuristics toward evidenced-based budget allocation thus allowing companies to achieve higher marketing efficiencies and better financial performance.

References

Nuara, M., & Alessandro, F. (2021). Data-driven marketing budget allocation: A machine learning approach. *Politecnico di Milano*.

IEEE Xplore. (2024). AI-powered marketing analytics: Optimizing budget allocation using predictive modeling.

EBSCO Research. (2024). Machine learning applications in marketing: Enhancing campaign efficiency through AI-driven strategies.

Generative Pre-trained Transformer (ChatGPT): For Model Coding

GitHub Repository Link

https://github.com/abhisheksharma2412/marketing_budget_project

Appendix

Appendix A: Visualization of data

Figure 1: Count of campaign and channel categories

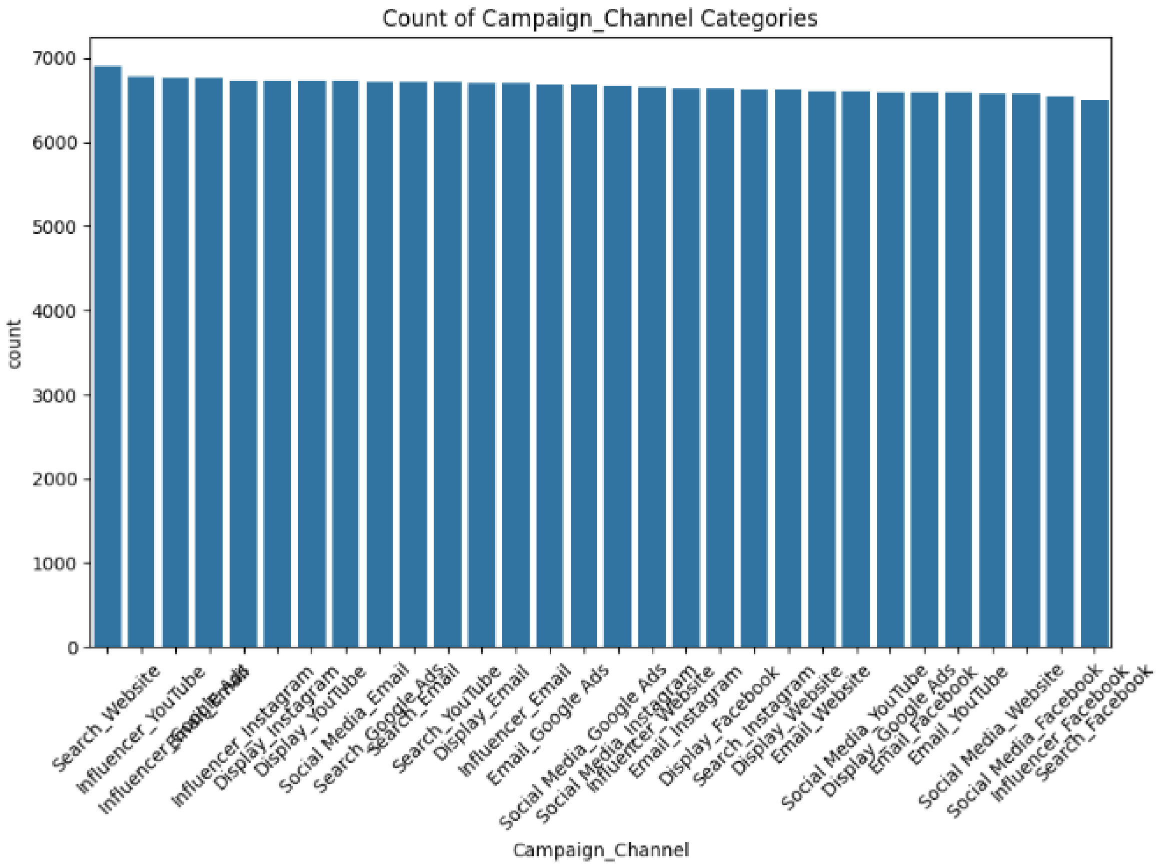


Figure 2: Average ROI by campaign type and channel used

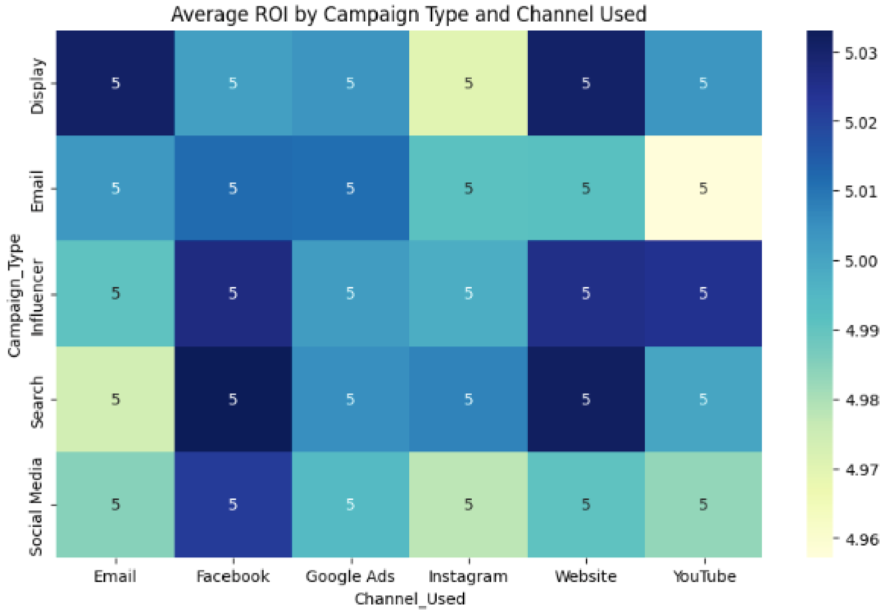


Figure 3: Frequency of Target Audiences

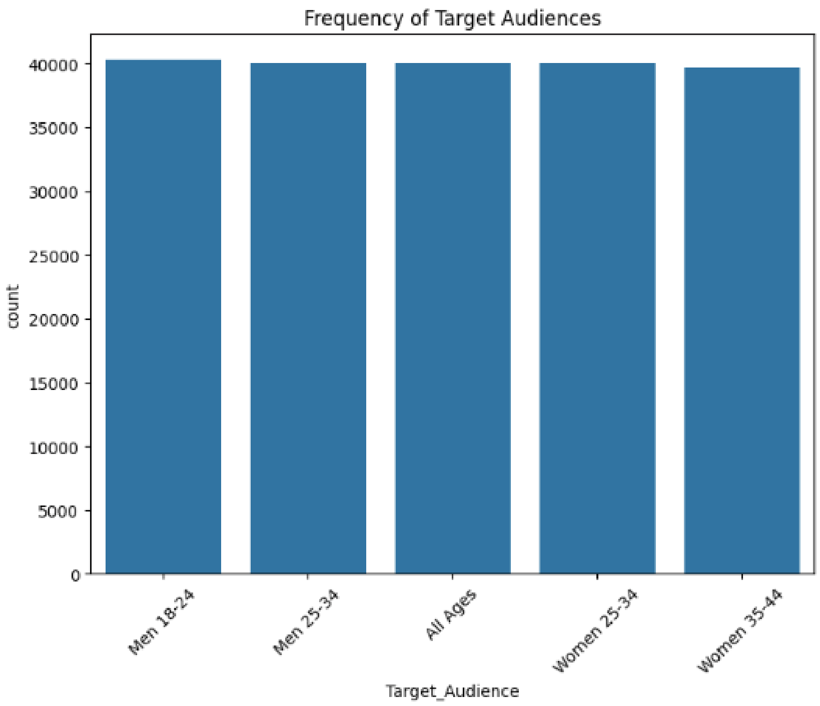


Figure 4: ROI Distribution by customer segment

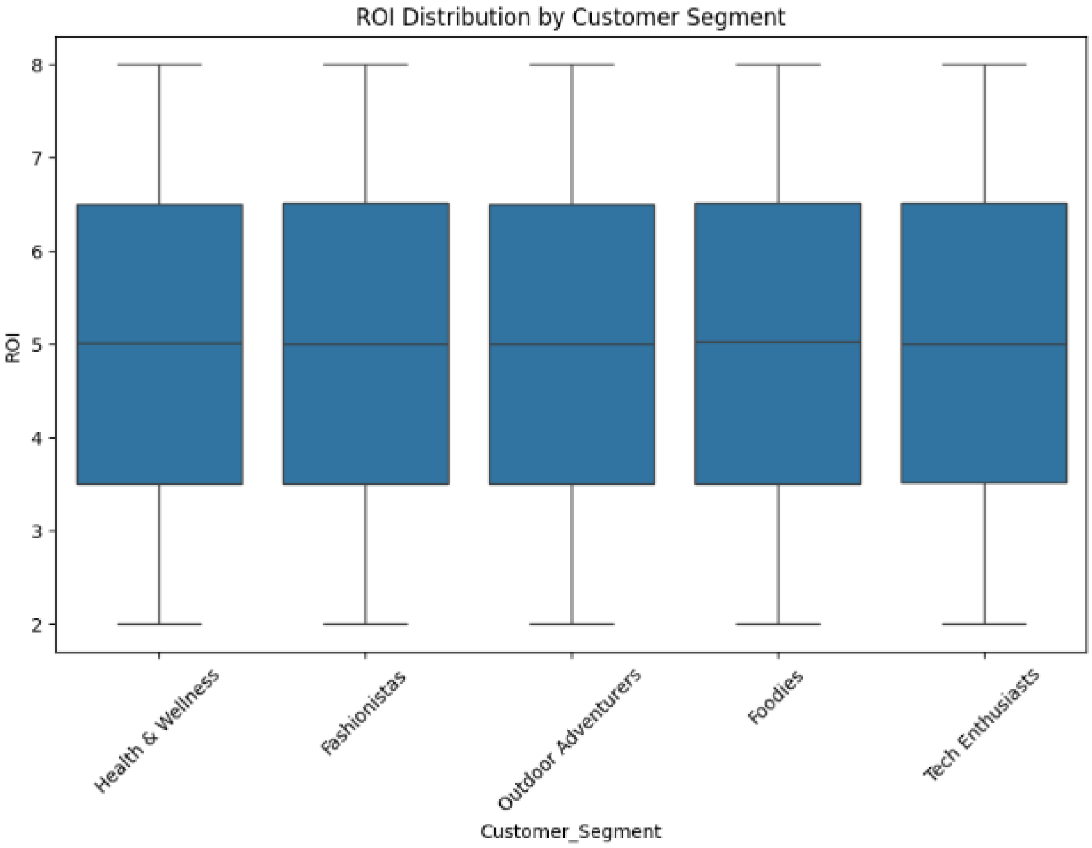


Figure 5:Distribution of ROI, Clicks, Impressions and Acquisition Cost

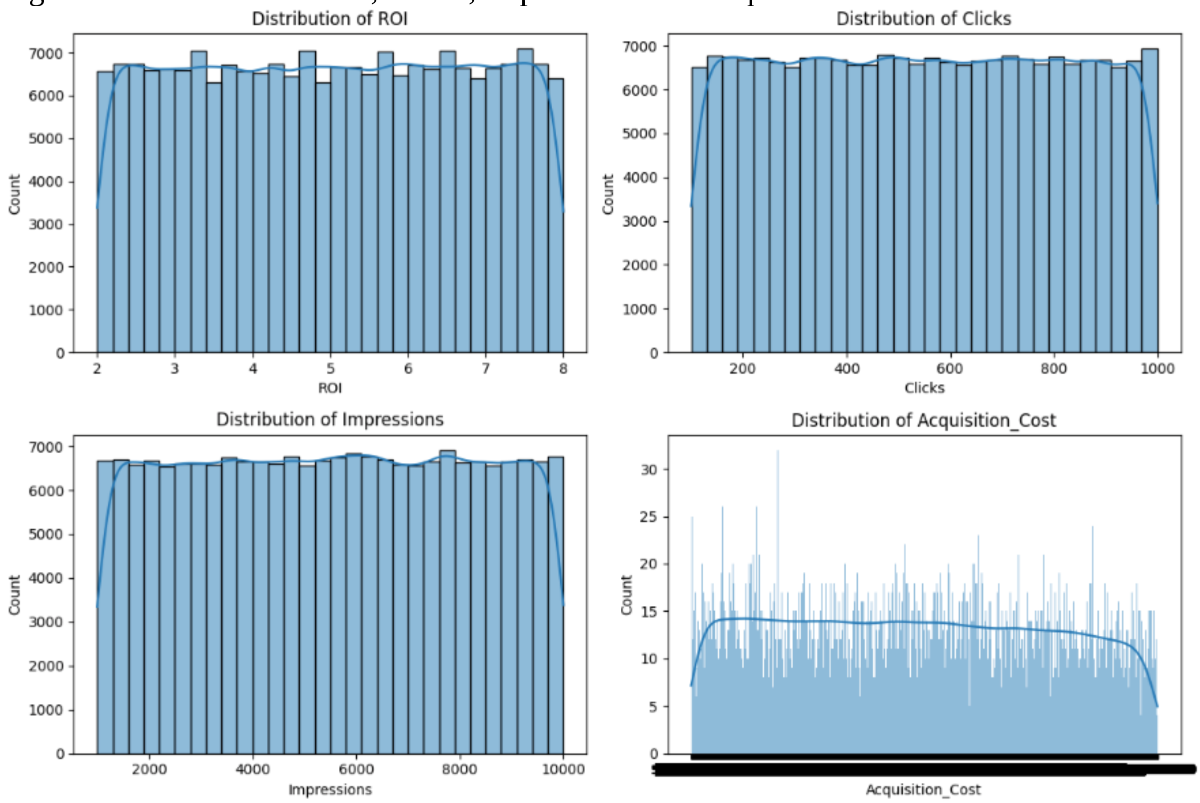


Figure 6: Correlation Matrix of Key Features

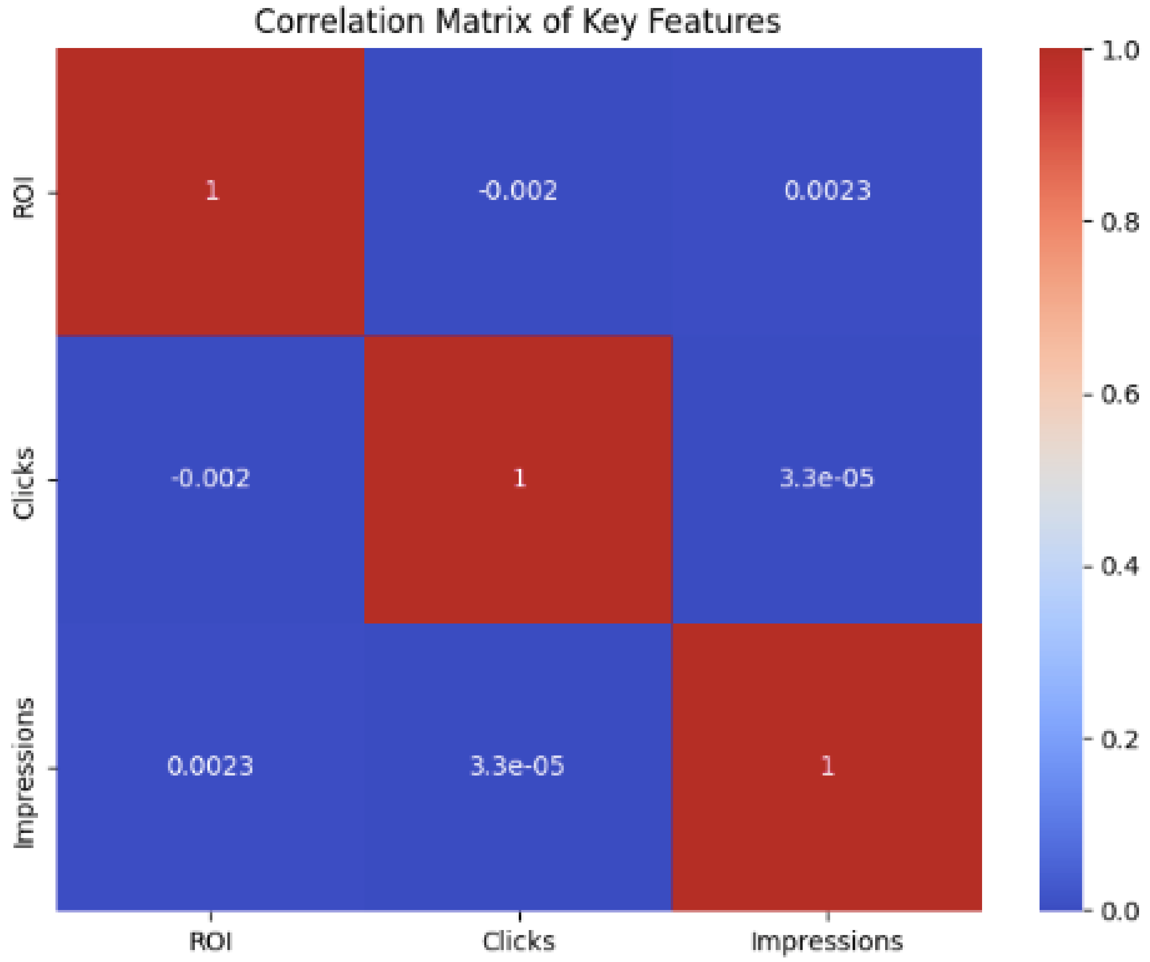
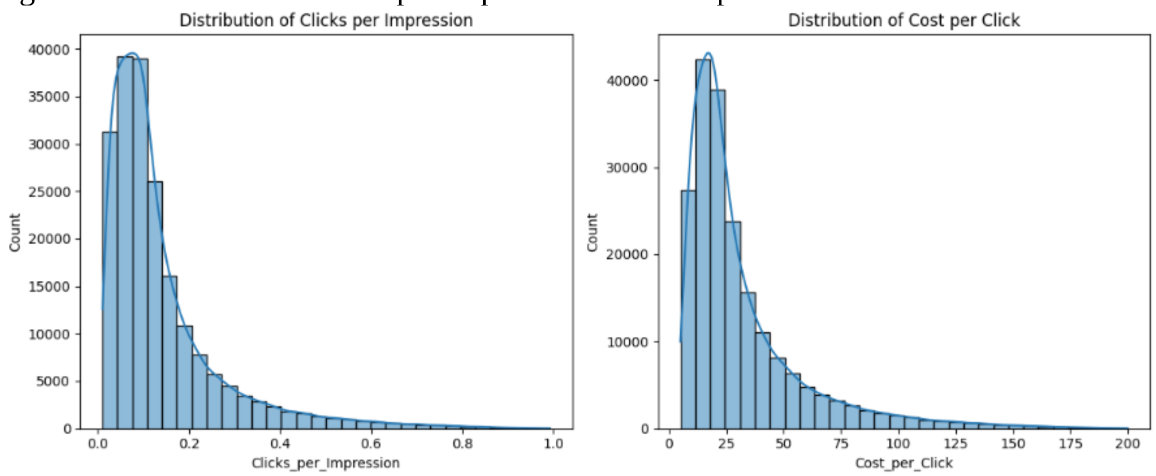


Figure 7:Distribution of Clicks per Impression and Cost per Click



Appendix B: Tabular representation

Table 1: Dataset Overview

First 5 rows of the dataset:

	Campaign_ID	Company	Campaign_Type	Target_Audience	Duration
0	1	Innovate Industries	Email	Men 18-24	30 days
1	2	NexGen Systems	Email	Women 35-44	60 days
2	3	Alpha Innovations	Influencer	Men 25-34	30 days
3	4	DataTech Solutions	Display	All Ages	60 days
4	5	NexGen Systems	Email	Men 25-34	15 days

	Channel_Used	Conversion_Rate	Acquisition_Cost	ROI	Location	Language
0	Google Ads	0.04	$16,174.00	6.29	Chicago	Spanish
1	Google Ads	0.12	$11,566.00	5.61	New York	German
2	YouTube	0.07	$10,200.00	7.18	Los Angeles	French
3	YouTube	0.11	$12,724.00	5.55	Miami	Mandarin
4	YouTube	0.05	$16,452.00	6.50	Los Angeles	Mandarin

	Clicks	Impressions	Engagement_Score	Customer_Segment	Date
0	506	1922	6	Health & Wellness	2021-01-01
1	116	7523	7	Fashionistas	2021-01-02
2	584	7698	1	Outdoor Adventurers	2021-01-03
3	217	1820	7	Health & Wellness	2021-01-04
4	379	4201	3	Health & Wellness	2021-01-05

Table 2: Statistics Summary

Summary Statistics:

	Campaign_ID	Conversion_Rate	ROI	Clicks
count	200000.000000	200000.000000	200000.000000	200000.000000
mean	100000.500000	0.080070	5.002438	549.772030
std	57735.171256	0.040602	1.734488	260.019056
min	1.000000	0.010000	2.000000	100.000000
25%	50000.750000	0.050000	3.500000	325.000000
50%	100000.500000	0.080000	5.010000	550.000000
75%	150000.250000	0.120000	6.510000	775.000000
max	200000.000000	0.150000	8.000000	1000.000000

	Impressions	Engagement_Score
count	200000.000000	200000.000000
mean	5507.301520	5.494710
std	2596.864286	2.872581
min	1000.000000	1.000000
25%	3266.000000	3.000000
50%	5517.500000	5.000000
75%	7753.000000	8.000000
max	10000.000000	10.000000

Table 3: New Tables after Feature Engineering

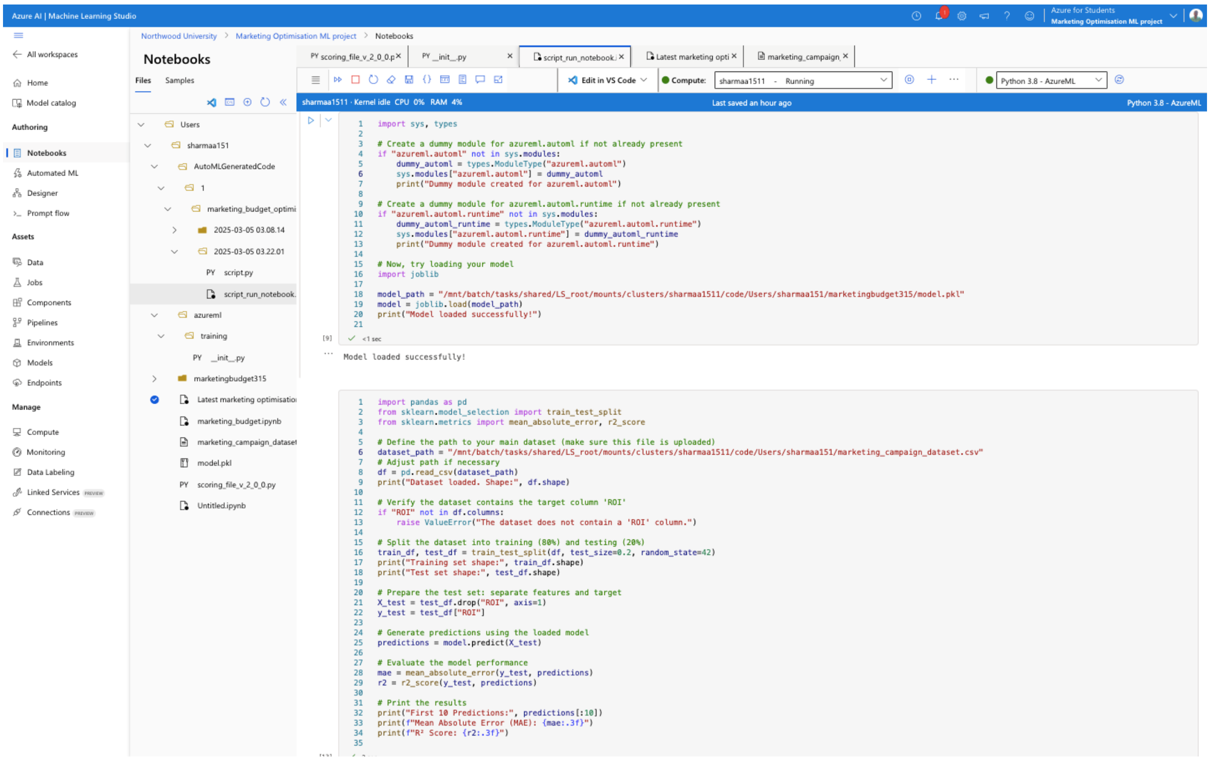
```
Dataset with New Features:
   Clicks  Impressions  Clicks_per_Impression  Acquisition_Cost  \
0     506         1922               0.263267           16174.0
1     116         7523               0.015419           11566.0
2     584         7698               0.075864           10200.0
3     217         1820               0.119231           12724.0
4     379         4201               0.090217           16452.0

   Cost_per_Click       Date  Month  Day  Quarter    Campaign_Channel
0       31.964427 2021-01-01      1    1        1    Email_Google Ads
1       99.706897 2021-01-02      1    2        1    Email_Google Ads
2       17.465753 2021-01-03      1    3        1  Influencer_YouTube
3       58.635945 2021-01-04      1    4        1     Display_YouTube
4       43.408971 2021-01-05      1    5        1       Email_YouTube
```

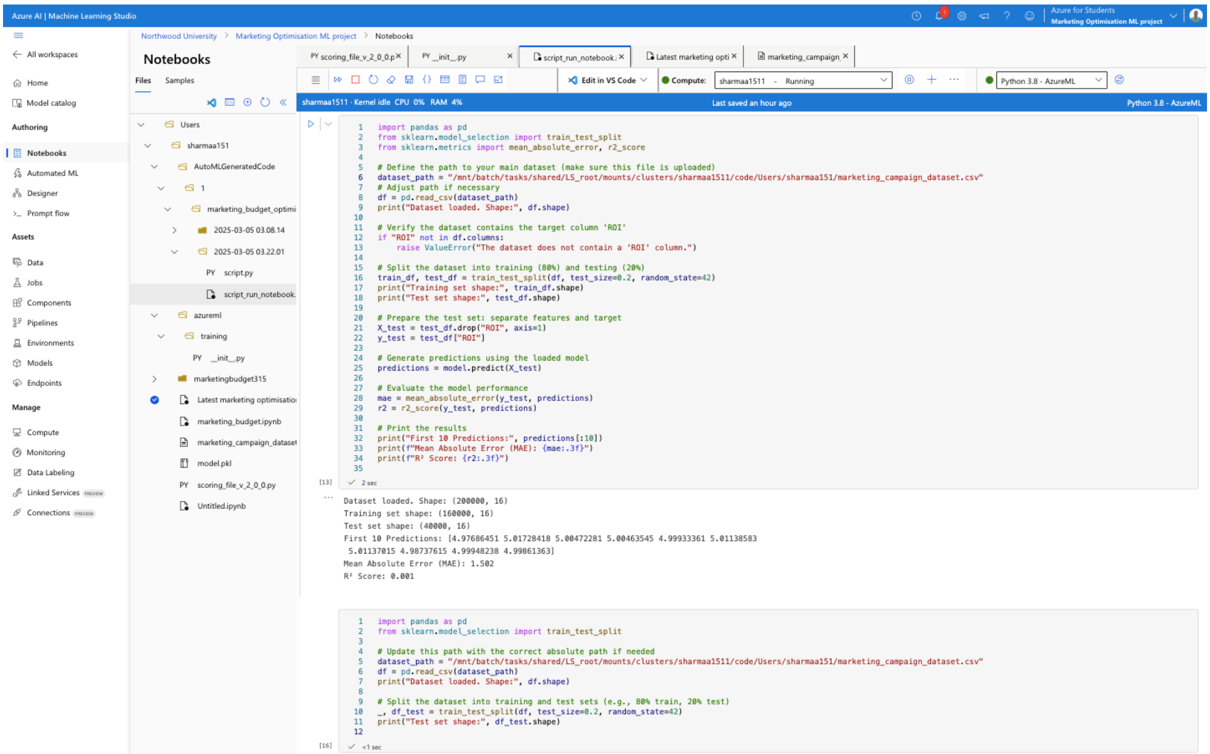
Table 4: Skewness of numerical figures and Dataset with Derived Features and Transformations

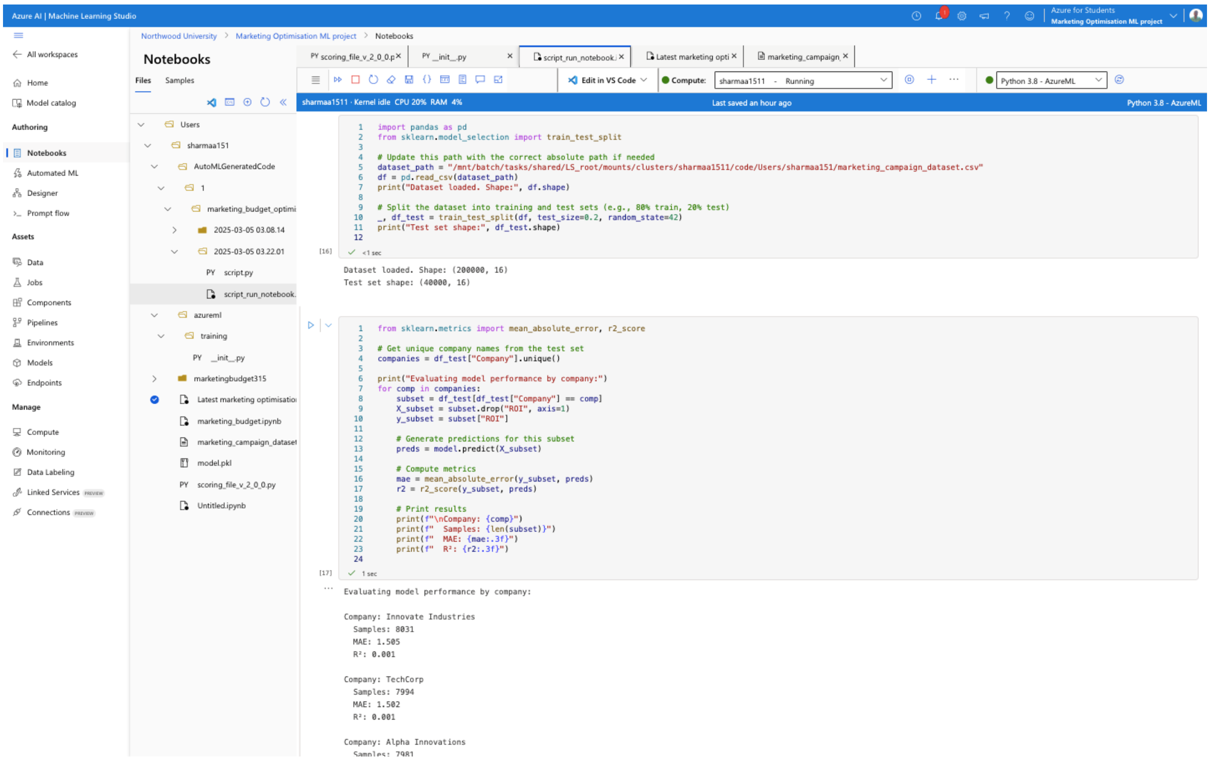
```
Skewness of numeric features:
Acquisition_Cost         0.001846
Clicks                   0.001423
Impressions             -0.004118
Cost_per_Click           2.209262
Clicks_per_Impression    2.315530
ROI                     -0.005043
dtype: float64
Applied log1p transformation to 'Cost_per_Click', created 'Cost_per_Click_log'.
Applied log1p transformation to 'Clicks_per_Impression', created 'Clicks_per_Impression_log'.

Dataset with Derived Features and Transformations (first 5 rows):
   Campaign_ID              Company Campaign_Type Target_Audience  Duration  \
0            1  Innovate Industries         Email       Men 18-24        30
1            2       NexGen Systems         Email     Women 35-44        60
2            3    Alpha Innovations    Influencer       Men 25-34        30
3            4   DataTech Solutions       Display        All Ages        60
4            5       NexGen Systems         Email       Men 25-34        15

  Channel_Used  Conversion_Rate  Acquisition_Cost   ROI     Location  ...  \
0   Google Ads             0.04           16174.0  6.29      Chicago  ...
1   Google Ads             0.12           11566.0  5.61     New York  ...
2      YouTube             0.07           10200.0  7.18  Los Angeles  ...
3      YouTube             0.11           12724.0  5.55        Miami  ...
4      YouTube             0.05           16452.0  6.50  Los Angeles  ...

  Cost_per_Click               Audience_Segment  Day_of_Week  Is_Weekend  \
0      31.964427    Men 18-24_Health & Wellness       Friday           0
1      99.706897      Women 35-44_Fashionistas     Saturday           1
2      17.465753  Men 25-34_Outdoor Adventurers       Sunday           1
3      58.635945     All Ages_Health & Wellness       Monday           0
4      43.408971    Men 25-34_Health & Wellness      Tuesday           0

  Month Day  Quarter  Is_Holiday Cost_per_Click_log Clicks_per_Impression_log
0     1   1        1           1           3.495429                  0.233702
1     1   2        1           0           4.612214                  0.015302
2     1   3        1           0           2.915918                  0.073124
3     1   4        1           0           4.088258                  0.112642
4     1   5        1           0           3.793441                  0.086376

[5 rows x 27 columns]
```

Appendix C: Model Development Code

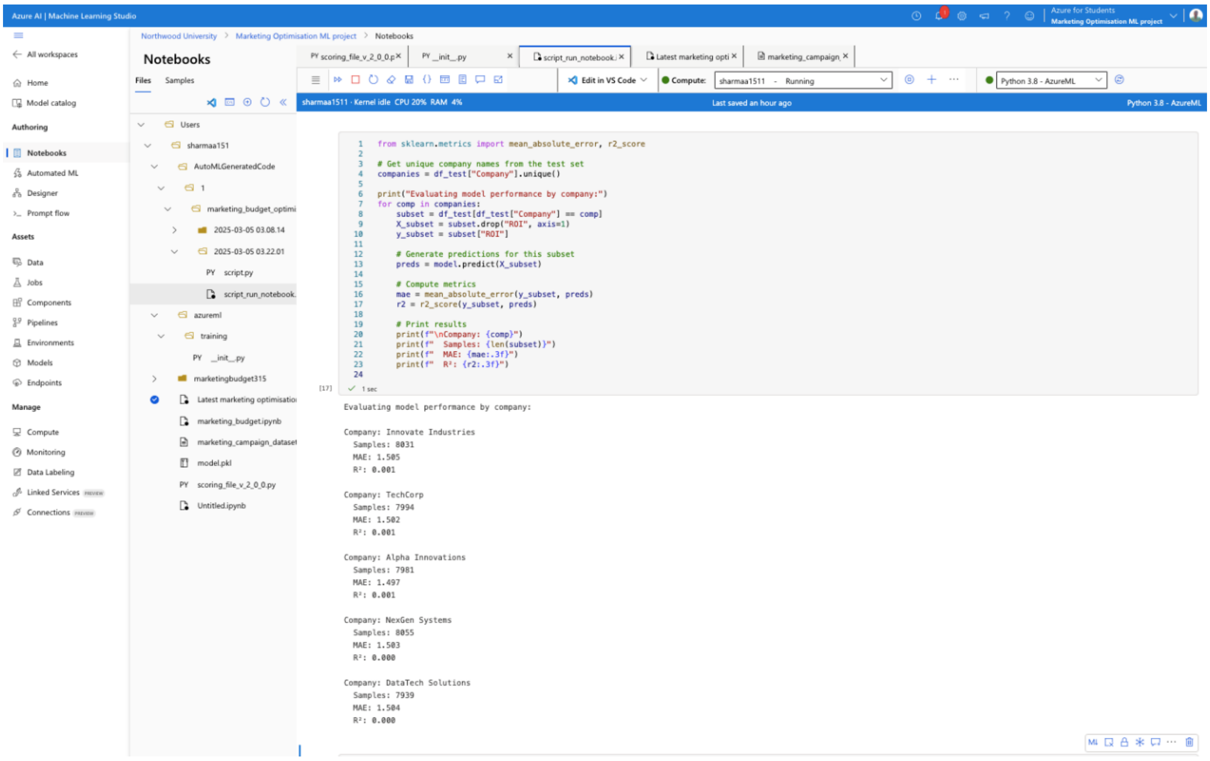
Code 1, 2 and 3: Deployment of Model


Azure AI | Machine Learning Studio
Northwood University > Marketing Optimisation ML project > Notebooks
Notebooks
import sys, types
# Create a dummy module for azureml.automl if not already present
if "azureml.automl" not in sys.modules:
dummy_automl = types.ModuleType("azureml.automl")
sys.modules["azureml.automl"] = dummy_automl
print("Dummy module created for azureml.automl")
# Create a dummy module for azureml.automl.runtime if not already present
if "azureml.automl.runtime" not in sys.modules:
dummy_automl_runtime = types.ModuleType("azureml.automl.runtime")
sys.modules["azureml.automl.runtime"] = dummy_automl_runtime
print("Dummy module created for azureml.automl.runtime")
# Now, try loading your model
import joblib
model_path = "/mnt/batch/tasks/shared/LS_root/mounts/clusters/sharmaa1511/code/Users/sharmaa151/marketingbudget315/model.pkl"
model = joblib.load(model_path)
print("Model loaded successfully!")
Model loaded successfully!
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score
# Define the path to your main dataset (make sure this file is uploaded)
dataset_path = "/mnt/batch/tasks/shared/LS_root/mounts/clusters/sharmaa1511/code/Users/sharmaa151/marketing_campaign_dataset.csv"
# Adjust path if necessary
df = pd.read_csv(dataset_path)
print("Dataset loaded. Shape:", df.shape)
# Verify the dataset contains the target column 'ROI'
if "ROI" not in df.columns:
raise ValueError("The dataset does not contain a 'ROI' column.")
Split the dataset into training (80%) and testing (20%)
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)
print("Training set shape:", train_df.shape)
print("Test set shape:", test_df.shape)
# Prepare the test set: separate features and target
X_test = test_df.drop("ROI", axis=1)
y_test = test_df["ROI"]
# Generate predictions using the loaded model
predictions = model.predict(X_test)
# Evaluate the model performance
mae = mean_absolute_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
# Print the results
print("First 10 Predictions:", predictions[:10])
print(f"Mean Absolute Error (MAE): {mae:.3f}")
print(f"R² Score: {r2:.3f}")

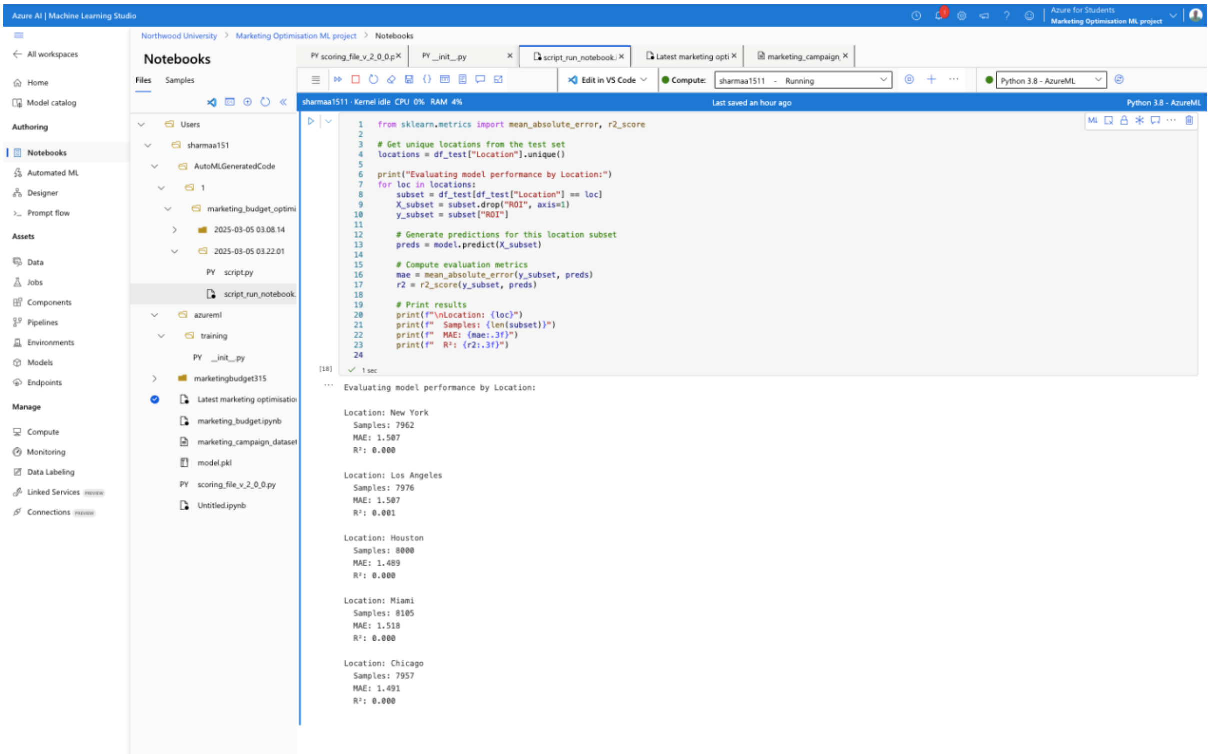

Azure AI | Machine Learning Studio
Northwood University > Marketing Optimisation ML project > Notebooks
Notebooks
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score
# Define the path to your main dataset (make sure this file is uploaded)
dataset_path = "/mnt/batch/tasks/shared/LS_root/mounts/clusters/sharmaa1511/code/Users/sharmaa151/marketing_campaign_dataset.csv"
# Adjust path if necessary
df = pd.read_csv(dataset_path)
print("Dataset loaded. Shape:", df.shape)
# Verify the dataset contains the target column 'ROI'
if "ROI" not in df.columns:
raise ValueError("The dataset does not contain a 'ROI' column.")
Split the dataset into training (80%) and testing (20%)
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)
print("Training set shape:", train_df.shape)
print("Test set shape:", test_df.shape)
# Prepare the test set: separate features and target
X_test = test_df.drop("ROI", axis=1)
y_test = test_df["ROI"]
# Generate predictions using the loaded model
predictions = model.predict(X_test)
# Evaluate the model performance
mae = mean_absolute_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
# Print the results
print("First 10 Predictions:", predictions[:10])
print(f"Mean Absolute Error (MAE): {mae:.3f}")
print(f"R² Score: {r2:.3f}")
Dataset loaded. Shape: (200000, 16)
Training set shape: (160000, 16)
Test set shape: (40000, 16)
First 10 Predictions: [4.97686451 5.01728418 5.00472281 5.00463545 4.99933361 5.01138583
5.01137015 4.98737615 4.99948238 4.99861363]
Mean Absolute Error (MAE): 1.502
R² Score: 0.001
import pandas as pd
from sklearn.model_selection import train_test_split
# Update this path with the correct absolute path if needed
dataset_path = "/mnt/batch/tasks/shared/LS_root/mounts/clusters/sharmaa1511/code/Users/sharmaa151/marketing_campaign_dataset.csv"
df = pd.read_csv(dataset_path)
print("Dataset loaded. Shape:", df.shape)
# Split the dataset into training and test sets (e.g., 80% train, 20% test)
_, df_test = train_test_split(df, test_size=0.2, random_state=42)
print("Test set shape:", df_test.shape)



Code 4: Evaluating Model by Company



Code 5: Evaluating Model by Location


Azure AI | Machine Learning Studio
Northwood University
Marketing Optimisation ML project
Notebooks
script_run_notebook
Edit in VS Code
Compute:
sharmaa1511 - Running
Python 3.8 - AzureML
sharmaa1511 · Kernel idle CPU 0% RAM 4%
Last saved an hour ago
from sklearn.metrics import mean_absolute_error, r2_score
# Get unique locations from the test set
locations = df_test["Location"].unique()
print("Evaluating model performance by Location:")
for loc in locations:
subset = df_test[df_test["Location"] == loc]
X_subset = subset.drop("ROI", axis=1)
y_subset = subset["ROI"]
# Generate predictions for this location subset
preds = model.predict(X_subset)
# Compute evaluation metrics
mae = mean_absolute_error(y_subset, preds)
r2 = r2_score(y_subset, preds)
# Print results
print(f"\nLocation: {loc}")
print(f" Samples: {len(subset)}")
print(f" MAE: {mae:.3f}")
print(f" R²: {r2:.3f}")
Evaluating model performance by Location:
Location: New York
Samples: 7962
MAE: 1.507
R²: 0.000
Location: Los Angeles
Samples: 7976
MAE: 1.507
R²: 0.001
Location: Houston
Samples: 8000
MAE: 1.489
R²: 0.000
Location: Miami
Samples: 8105
MAE: 1.518
R²: 0.000
Location: Chicago
Samples: 7957
MAE: 1.491
R²: 0.000