GOMYCODE

# MACHINE LEARNING PROJECT

## E-Commerce Ad Spend Optimizer ML Model

**Glory Archibong**
Data Science and Analysis Bootcamp
Machine Learning (DNN) Checkpoint
10th Dec 2024

# Contents

## INTRODUCTION

Optimizing advertising spend is crucial for maximizing return on investment (ROI) and driving business growth. However, with multiple advertising channels, complex customer behaviors, and limited budgets, allocating resources effectively is challenging. The primary objective of this project is to develop a machine learning model that optimizes e-commerce ad spend by predicting the return on ad spend (ROAS) based on various factors such as customer demographics, product categories, and advertising metrics.

### Dataset Overview

The dataset used for this project is a synthetic e-commerce dataset that mimics real-world scenarios. The dataset contains 100,000 transactions with 14 features, including transaction ID, customer ID, product ID, transaction date, units sold, discount applied, revenue, clicks, impressions, conversion rate, category, region, ad CTR, and ad CPC.

### Significance

The project has significant real-world applications, as optimizing ad spend can help e-commerce businesses increase their revenue and improve their return on investment (ROI).

## DATASET SELECTION AND VISUALIZATION

The synthetic e-commerce dataset was chosen for this project because it is relevant to the task at hand and can mimic real-world scenarios.

### Task Definition

The task is a regression problem, where the goal is to predict the ROAS based on the given features.

### Initial Exploration

The dataset was explored using various techniques to understand the distribution of the data and the relationships between the features.

This included:

*Data Profiling*

The ydata profiling report provided a comprehensive overview of the dataset, including summary statistics, distribution plots, and correlation matrices.

*Univariate Analysis*

Bar charts and histograms were used to visualize the distribution of the individual features, including the target variable ROAS. Some of which are:

- Bar Chart of the relatively independent variables - 'Category', 'Region', 'Product_ID'
- Histogram visuals for each feature to see the data distribution.

*# Multivariate Analysis*

Scatter plots and boxplots were used to visualize the relationships between the features and the target variable. Such as:

- The use of Matplotlib to create boxplots for the numerical variables to visualize their outliers.
- Creating a Scatter Plot to visualize the relationship between Average Order Value (AOV) and Revenue, using Seaborn
- Creating a Scatter Plot to visualize the relationship between Ad Spend and Return on Ad Spend, using Seaborn
- Creating line charts to visualize the monthly revenue trend and Ad Spend over time
- Creating a line chart for Ad Spend Over Time to examine the Ad Spend trend over time.

# DATA PREPROCESSING

## Missing Value Handling

There were no missing values in the dataset.

## Data Cleaning

The data was cleaned by converting the transaction date feature to a datetime format and handling inconsistencies in the category and region features.

## Data Transformation

The data was transformed by scaling the numerical features using the StandardScaler from scikit-learn and encoding the categorical features using one-hot encoding.

## FEATURE ENGINEERING

### Feature Creation

Two new features were created: average order value (AOV) and return on ad spend (ROAS). AOV was calculated by dividing the revenue by the number of units sold, and ROAS was calculated by dividing the revenue by the ad spend.

### Feature Transformation

The features were transformed by applying logarithmic scaling to the AOV and ROAS features.

### Feature Selection

The features were selected using correlation analysis and recursive feature elimination. The results showed that the AOV, ROAS, and ad CTR features were the most important for predicting the ROAS while the irrelevant variables like Transaction_ID, Customer_ID, and Transaction_Date were excluded from the model training. Analysis of the Correlation Matrix obtained from the Ydata profile revealed,

*Strong Positive Correlations*

- AOV and Revenue: Highly correlated (0.908), indicating that Average Order Value is a strong driver of Revenue.
- ROAS and Revenue: Strongly correlated (0.669), suggesting that Return on Ad Spend is a key factor in driving Revenue.
- Clicks and Conversion_Rate: Highly correlated (0.697), indicating that Clicks are a strong predictor of Conversion Rates.

*Strong Negative Correlations*

- Ad_Spend and ROAS: Strongly negatively correlated (-0.700), suggesting that increasing Ad Spend may not always lead to higher Return on Ad Spend.
- AOV and Units_Sold: Negatively correlated (-0.381), indicating that higher Average Order Values may be associated with lower Units Sold.

*Weak Correlations*

- Category and other variables: Most correlations with Category are weak, suggesting that Category may not be a strong driver of other variables.
- Region and other variables: Most correlations with Region are weak, indicating that Region may not have a significant impact on other variables.

## MODEL SELECTION

### Model Choice

Several machine learning models were evaluated for their ability to predict the ROAS variable. The models considered were:

- Linear Regression: A linear regression model was used as a baseline to evaluate the performance of the other models.
- Decision Tree Regressor: A decision tree regressor was used to evaluate the performance of a simple tree-based model.
- Random Forest Regressor: A random forest regressor was used to evaluate the performance of an ensemble tree-based model.
- Gradient Boosting Regressor: A gradient boosting regressor was used to evaluate the performance of another ensemble tree-based model.

### Algorithm Explanation

The Random Forest Regressor works by creating multiple decision trees and combining their predictions to produce a final output. This approach helps to reduce overfitting and improve the overall performance of the model.

The Random Forest Regressor was selected as the final model due to its high performance and ability to handle the complexity of the data.

## MODEL EVALUATION

The performance of each model was evaluated using the following metrics:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Coefficient of Determination (R2)
- Mean Absolute Percentage Error (MAPE)

### Performance Results

The results of the model evaluation are presented below:

Linear Regression Performance Metrics:
- MSE: 643.57
- RMSE: 25.37
- MAE: 11.98
- R2: 0.38
- MAPE: 512.64%

Random Forest Regressor Performance Metrics:
- MSE: 4.77
- RMSE: 2.18
- MAE: 0.16
- R2: 1.00
- MAPE: 0.55%

Decision Tree Performance Metrics:
- MSE: 10.92
- RMSE: 3.30
- MAE: 0.41
- R2: 0.99
- MAPE: 1.74%

Gradient Boosting Regressor Performance Metrics:
- MSE: 5.52
- RMSE: 2.35
- MAE: 1.14
- R2: 0.99
- MAPE: 68.11%

**MODEL IMPROVEMENT**

The model was improved by using the cross-validation technique to evaluate its performance. The results are presented below:

Linear Regression Cross-Validation Scores:

- MSE: 606.09
- MAE: 11.96
- R2: 0.39

Random Forest Cross-Validation Scores:

- MSE: 8.76
- MAE: 0.19
- R2: 0.99

Decision Trees Cross-Validation Scores:

- MSE: 24.30
- MAE: 0.48
- R2: 0.98

Gradient Boosting Cross-Validation Scores:

- MSE: 5.36
- MAE: 0.98
- R2: 0.99

Based on the results, the Random Forest Regressor model outperformed the other models, with the lowest MSE, RMSE, and MAE, and the highest R2**.**

**MODEL DEPLOYMENT**

The model was deployed using Streamlit, a Python library that allows users to create web applications.

*User Interface (UI)*

The UI was designed to allow users to input their data and receive predictions from the model.

*Usage Guide*

To use the app, users simply need to input their data into the UI and click the "Predict" button.

## CHALLENGES AND LEARNINGS

During the project, several challenges were encountered:

- Handling High-Dimensional Data: The dataset had a large number of features, which made it challenging to select the most relevant features for the model.
- Dealing with Skewed Data Distributions: Some of the features had skewed distributions, which required careful preprocessing to ensure that the model was not biased towards certain values.
- Handling Mixed Data Types: The dataset contained a mix of numerical and categorical features, which required careful encoding and preprocessing to ensure that the models could process them correctly.
- Model Overfitting: Some of the models, especially the decision tree and gradient boosting models, suffered from overfitting, which required careful tuning of hyperparameters to prevent.

Key Learnings

The project provided several key learnings:

- Importance of Feature Engineering: The project highlighted the importance of feature engineering in machine learning. The creation of new features, such as the average order value and return on ad spend, significantly improved the performance of the models.
- Need for Model Tuning: The project demonstrated the need for careful model tuning to prevent overfitting and ensure optimal performance.
- Value of Cross-Validation: The project showed the value of cross-validation in evaluating model performance and preventing overfitting.

## CONCLUSION AND REFERENCES

### Conclusion

This project demonstrated the use of machine learning to optimize e-commerce ad spend. The results showed that the model was able to accurately predict the ROAS based on the given features. The project also highlighted the importance of feature engineering and selection in machine learning.

### References:

**- Synthetic e-commerce dataset (source: https://www.kaggle.com/datasets/imranalishahh/comprehensive-synthetic-e-commerce-dataset.)**