Marketing Response Prediction for Retail Campaigns

Almog Cohen

Project Overview:

This project focuses on developing a **response model** to enhance the efficiency of marketing campaigns for a retail company. The dataset, sourced from Kaggle, contains detailed customer profiles, including demographics, past campaign responses, spending behavior, and online activity.

The primary objective is to **predict whether a customer will respond positively to the next marketing offer**. By accurately identifying likely responders, the company can **optimize campaign targeting**, reduce costs, and increase return on investment.

Business Problem:

Traditional mass marketing campaigns often suffer from low response rates and high expenses. To improve marketing ROI, companies need a way to target the **right customers**—those most likely to accept promotional offers.

Using historical data about customer interactions, complaints, and past campaign responses, this project aims to:

- **Classify** whether a customer will respond to the next campaign.
- Enable **data-driven targeting** to improve campaign precision.
- Support **profit maximization** by reducing wasteful outreach and boosting response efficiency.

Dataset Summary:

The dataset contains information on approximately 2,200 customers of a **retail company**, likely in the **supermarket or consumer goods sector**. Each row represents a unique customer, and the features span several key categories:

- **Demographics** (e.g., education, marital status, income, household size)
- **Purchase behavior** (e.g., amount spent on different product categories)
- **Channel interaction** (e.g., purchases via web, store, catalog)
- **Campaign history** (e.g., responses to past campaigns, complaints)
- **Recency and engagement** (e.g., days since last purchase, web visits)

The **target variable** is Response, indicating whether a customer accepted the **most recent marketing offer**.

This rich dataset enables building a predictive model that identifies which customers are more likely to respond positively to future campaigns.

Project Pipeline:

The project followed a structured data science workflow to ensure clarity, reproducibility, and performance. Below are the key steps:

1. **Data Preparation**
    - Removed irrelevant columns (e.g., ID)
    - Treated special values (DtCustomer, Income, Kidhome, Teenhome) and recoded where necessary
    - Converted categorical features using encoding techniques (One-Hot and Ordinal Encoding)
2. **Exploratory Data Analysis (EDA)**
    - Grouped features into numeric, categorical, and binary categories
    - Used visualizations (histograms, count plots) to uncover distributions and patterns
    - Investigated relationships between features and the target variable (Response)
3. **Feature Engineering**
    - Created new features based on domain logic and exploratory findings
    - Handled outliers using IQR and correlation analysis
    - Finalized a set of relevant features for modeling
4. **Feature Selection**
    - Used multiple models (Lasso, Ridge, Random Forest, Gradient Boosting, XGBoost) to rank features
    - Retained features with the highest average importance scores across models
5. **Model Selection & Evaluation**
    - Trained and compared popular classification models: Logistic Regression, SVM, Random Forest, Gradient Boosting, KNN, and XGBoost
    - Used AUC, F1 Score, and Confusion Matrix as main evaluation metrics
    - Assessed overfitting by comparing train/test AUC scores
6. **Model Fine-Tuning**
    - Applied RandomizedSearchCV for hyperparameter tuning on Logistic Regression and SVM
    - Final model selected: **Logistic Regression**, due to its balance of simplicity, interpretability, and performance
7. **Insights & Feature Importance**
    - Analyzed feature importance to understand what drives customer responses
    - Shared recommendations based on key influential features

Data Preparation Summary:

During the data preparation phase, the following actions were taken to ensure a clean and model-ready dataset:

- **Dropped Z_CostContact and Z_Revenue**: These columns had only a single unique value and thus provided no useful information for modeling.
- **Removed the ID column**: As it served only as a unique identifier, it had no predictive value.
- **Processed Dt_Customer**: The customer enrollment date was split into **year**, **month**, and **day** to extract potentially useful time-based features.
- **Converted data types**: Columns mistakenly labeled as object (text) were converted to appropriate types (e.g., int, float, or datetime).

Although the dataset was relatively clean, these small adjustments ensured consistent formatting and better compatibility with machine learning workflows.

Exploratory Data Analysis (EDA):

During the EDA phase, I explored the dataset to better understand feature behavior, relationships, and data quality. Key steps included:

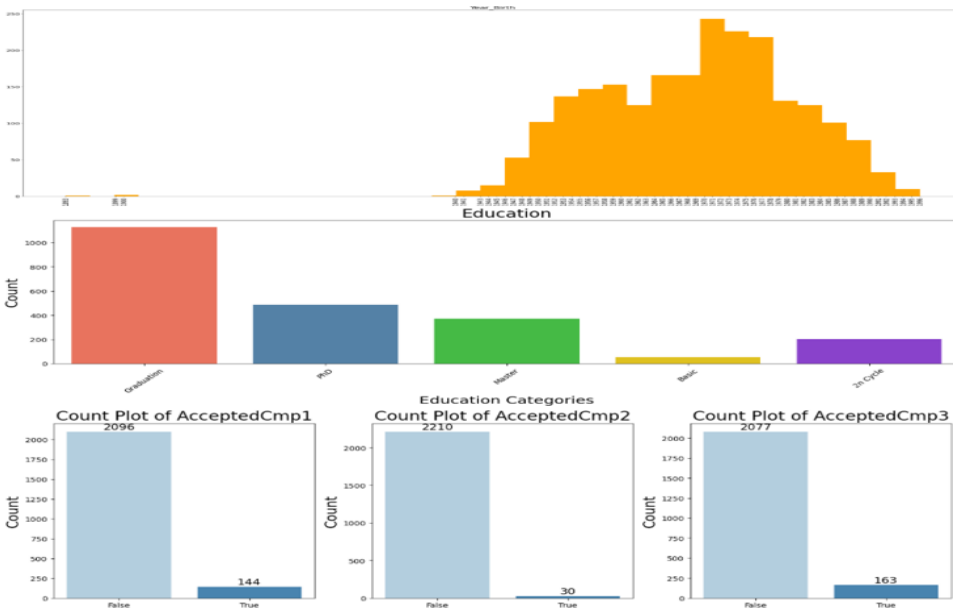
Feature Categorization

Features were grouped into three types to guide visualization and analysis:

- Numeric features (e.g., Income, Recency, NumWebPurchases)
- Categorical features (e.g., Education, Marital_Status)
- Dummy features (e.g., Complain, Kidhome, Teenhome)

Data Visualization

- Histograms were used to assess distributions of numeric features
- Count plots were used for categorical and dummy variables



Skewness Analysis

Skewness was examined for all numeric features:

- 6 features were approximately normal
- 1 feature was moderately skewed
- 7 features were highly skewed

Due to the skewed distribution, Spearman correlation (which does not assume linearity) was used for the next step.

Spearman Correlation Analysis (for Numeric Features)

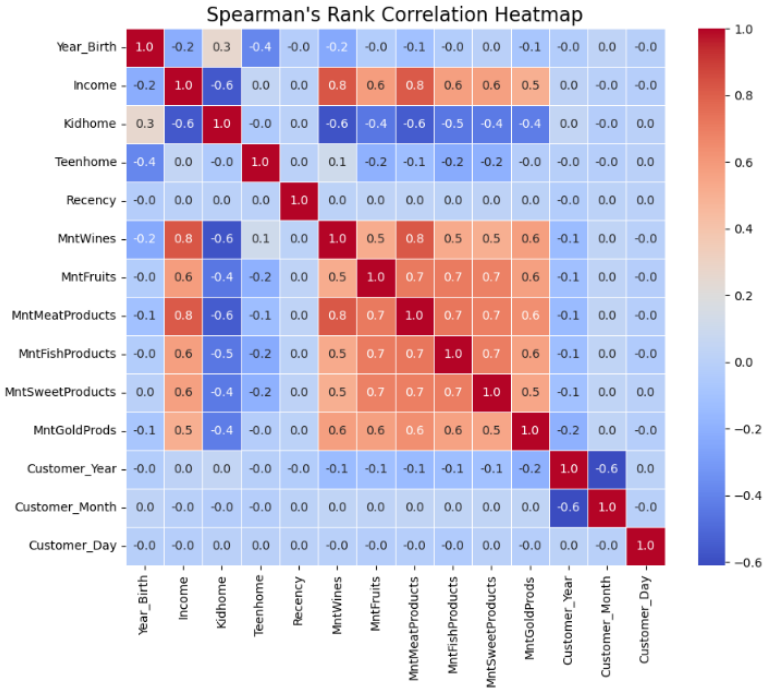
This method helped identify monotonic relationships:

Key Positive Correlations:

- Income & MntWines (0.83): Wealthier customers buy more wine
- Income & MntMeatProducts (0.82): Higher income → more meat spending
- MntWines & MntMeatProducts (0.82): Suggests a lifestyle pattern among customers
- Strong cluster among MntFishProducts, MntFruits, MntSweetProducts, MntGoldProds: Reflects “high-value” spenders

Key Negative Correlations:

- Kidhome & Income (-0.56): More kids at home often → lower income
- Kidhome & spending features (-0.58 to -0.42): Families with kids tend to spend less on premium items
- Year_Birth & Teenhome (-0.39): Older customers = fewer teenagers at home



Chi-Square Test (for Categorical & Dummy Features)

This test evaluated the statistical independence of categorical features with the target (Response):

- Strong associations:
  - NumWebPurchases, NumStorePurchases, NumCatalogPurchases, Education, Marital_Status
- Weak or no association:
  - Complain

Outlier Review

Outliers were detected using IQR analysis, and their validity was assessed feature by feature:

- Not removed:
  - MntFruits, MntMeatProducts, MntFishProducts, MntSweetProducts, MntGoldProds
    - These reflect low or zero spending and may be linked to personal behavior or preferences.
    - Not considered data errors.
  - Customer_Year
    - Outlier values like 2012 and 2014 are meaningful (based on time of acquisition).

- Removing them would eliminate valid, time-sensitive information.

Data Imputation

- Missing values in Income (24 instances) were filled using the median, to maintain robustness against skewed distribution.

Feature Engineering:

In this stage, the goal was to transform raw features into a format more suitable for machine learning. The following steps were applied:

- **New Feature Creation:**
  - **Total_Children**: Combined the number of kids (Kidhome) and teenagers (Teenhome) into a single feature.
  - **Total_Spent**: Aggregated the amount spent on all product categories — wine, fruits, meat, fish, sweets, and gold — into one comprehensive spending feature.
  - **Spending Ratios**: Created new ratio-based features to normalize product spending relative to total spending (e.g., Wine-to-Total-Spent Ratio).

Total_Children	Total_Spent	MntWines_Ratio	MntFruits_Ratio	MntMeatProducts_Ratio	MntFishProducts_Ratio	MntSweetProducts_Ratio	MntGoldProds_Ratio
0	1617	0.392703	0.054422	0.337662	0.106370	0.054422	0.054422
2	27	0.407407	0.037037	0.222222	0.074074	0.037037	0.222222
0	776	0.548969	0.063144	0.163660	0.143041	0.027062	0.054124
1	53	0.207547	0.075472	0.377358	0.188679	0.056604	0.094340
1	422	0.409953	0.101896	0.279621	0.109005	0.063981	0.035545

- **Encoding**
  - **Categorical Encoding**: Transformed string-based categorical features into numeric form using suitable encoding techniques (e.g., ordinal, one-hot).
  - **Boolean Encoding**: Converted binary dummy variables (e.g., Complain, Response) into 0/1 format to ensure compatibility with machine learning models.

Feature Selection:

To identify the most informative features for modeling, I applied multiple feature selection techniques, each offering a unique perspective on feature relevance. These included:

- **Lasso Regression** – a regularized linear model that penalizes less important features.
- **Ridge Regression** – similar to Lasso but less aggressive with coefficients.
- **Random Forest** – a tree-based ensemble that measures feature importance via impurity reduction.
- **Gradient Boosting** – builds additive models in a forward stage-wise fashion.
- **XGBoost** – an optimized gradient boosting technique well-suited for classification tasks.

Scoring Method:

Each model "voted" on feature importance. For every model that ranked a feature among the top predictors, that feature received one point. The total score for each feature ranged from 0 to 5.

	Feature	Lasso	GradientBoost	RandomForest	Ridge	XGBoost	Sum
0	Year_Birth	0	1	1	1	1	4
1	Education	1	1	1	1	1	5
2	Marital_Status	0	1	1	1	1	4
3	Income	0	1	1	1	1	4
4	Kidhome	0	0	1	1	1	3
5	Teenhome	1	1	1	1	1	5
6	Recency	1	1	1	1	1	5
7	MntWines	0	1	1	1	1	4
8	MntFruits	0	1	1	1	1	4
9	MntMeatProducts	1	1	1	1	1	5
10	MntFishProducts	0	1	1	1	1	4
11	MntSweetProducts	0	1	1	1	1	4
12	MntGoldProds	0	1	1	1	1	4
13	NumDealsPurchases	1	1	1	1	1	5
14	NumWebPurchases	1	1	1	1	1	5
15	NumCatalogPurchases	1	1	1	1	1	5
16	NumStorePurchases	1	1	1	1	1	5
17	NumWebVisitsMonth	1	1	1	1	1	5
18	AcceptedCmp3	1	1	1	1	1	5
19	AcceptedCmp4	1	1	1	1	1	5
20	AcceptedCmp5	1	1	1	1	1	5
21	AcceptedCmp1	1	1	1	1	1	5
22	AcceptedCmp2	1	1	1	1	1	5
23	Complain	0	0	1	1	1	3
24	Customer_Year	1	1	1	1	1	5
25	Customer_Month	1	1	1	1	1	5
26	Customer_Day	0	1	1	1	1	4
27	Total_Children	0	1	1	1	1	4
28	Total_Spent	1	1	1	1	1	5
29	MntWines_Ratio	0	1	1	1	1	4
30	MntFruits_Ratio	0	1	1	1	1	4
31	MntMeatProducts_Ratio	1	1	1	1	1	5
32	MntFishProducts_Ratio	1	1	1	1	1	5
33	MntSweetProducts_Ratio	0	1	1	1	1	4
34	MntGoldProds_Ratio	1	1	1	1	1	5

Feature Elimination Criteria:

- Features with a **score of 3 or below** were removed.
- Features with a **score of 4** were also removed after observing that Lasso (a strict regularizer) consistently excluded them, indicating weaker predictive power across models.
- Final decisions were influenced more heavily by **Random Forest** and **XGBoost**, due to their robustness in handling non-linear relationships and interactions in classification tasks.

This approach ensured that only the most consistently important features were retained, improving model simplicity and reducing noise in the data.

Model Selection:

At this stage, multiple classification algorithms were tested to identify the best-performing model for predicting customer response to marketing campaigns.

Models Trained:

- **Logistic Regression**
- **K-Nearest Neighbors (KNN)**
- **Decision Tree**
- **Random Forest**
- **XGBoost Classifier**

Each model was evaluated using training, validation, and test sets to avoid overfitting and ensure generalization

Evaluation Metrics:

For each model, the following performance metrics were recorded:

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**
- **ROC-AUC Score**

Conclusion: Evaluating Model Performance and Overfitting

All of the models in our evaluation demonstrated high performance metrics on the test set, but it is crucial to assess the possibility of overfitting, particularly when a model performs exceptionally well on the training set but shows a significant drop in performance on the test set.To detect overfitting, we:

1. Evaluated the AUC (Area Under the Curve) on both the training and test sets for all models.
2. Compared the AUC scores from the training set with those from the test set:
    - If a model exhibited a significantly higher AUC on the training set compared to the test set, it suggested potential overfitting.

Observations:

Logistic Regression and SVM:

Both models showed minimal differences between Train AUC and Test AUC, indicating good generalization:

- Logistic Regression: Train AUC = 0.6825, Test AUC = 0.6814 (Δ = 0.0011)
- SVM: Train AUC = 0.6919, Test AUC = 0.6657 (Δ = 0.0262)

Random Forest, Gradient Boosting, KNN, and XGBoost:

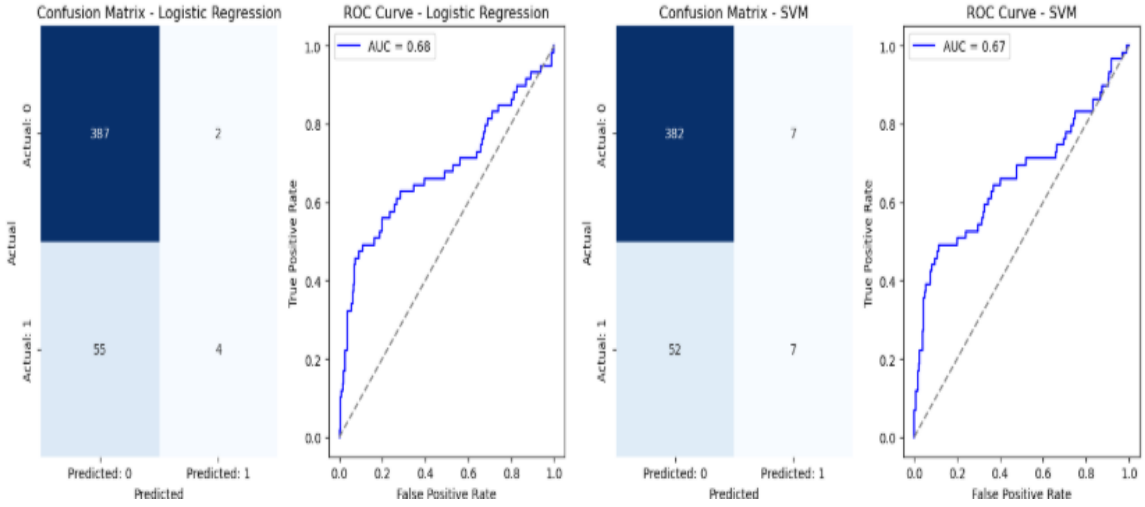
These models exhibited larger AUC gaps, signaling potential overfitting:

- Random Forest: Train AUC = 0.9997, Test AUC = 0.7096 (Δ = 0.2901)
- Gradient Boosting: Train AUC = 0.9078, Test AUC = 0.6669 (Δ = 0.2409)
- KNN: Train AUC = 0.8894, Test AUC = 0.6562 (Δ = 0.2332)
- XGBoost: Train AUC = 0.9886, Test AUC = 0.6705 (Δ = 0.3181)

Conclusion:

- Logistic Regression and SVM demonstrated the most reliable performance across both the training and test sets, indicating good generalization and minimal risk of overfitting. These models were selected for further fine-tuning.
- Tree-based models (Random Forest, Gradient Boosting, XGBoost) and KNN showed signs of overfitting, as their performance on the training set was significantly higher than on the test set. These models may still be useful with hyperparameter tuning or regularization to mitigate overfitting.
- By conducting this analysis, we ensured that our top models not only performed well on the training data but also generalized effectively to unseen data.

So the model chosen to be fine tuned are: SVM & Logistic regression



Final Model Selection and Fine-Tuning

Hyperparameter Optimization Using RandomizedSearchCV

After identifying **Logistic Regression** and **SVM** as the most promising models due to their strong performance and minimal overfitting, we proceeded to **fine-tune** both models using **RandomizedSearchCV**.

Why RandomizedSearchCV?

We chose RandomizedSearchCV over GridSearchCV for the following reasons:

- **Efficiency**: It samples from the hyperparameter space rather than searching exhaustively, which is faster and more scalable.
- **Flexibility**: It allows testing a broader range of hyperparameters even with limited computational resources.

Fine-Tuning Process

- **Logistic Regression - Tuned Parameters**: C, solver, max_iter
- **SVM - Tuned Parameters**: C, gamma, kernel

We applied cross-validation during the tuning process and selected the best models based on **AUC score**.

Model Chosen: Logistic Regression

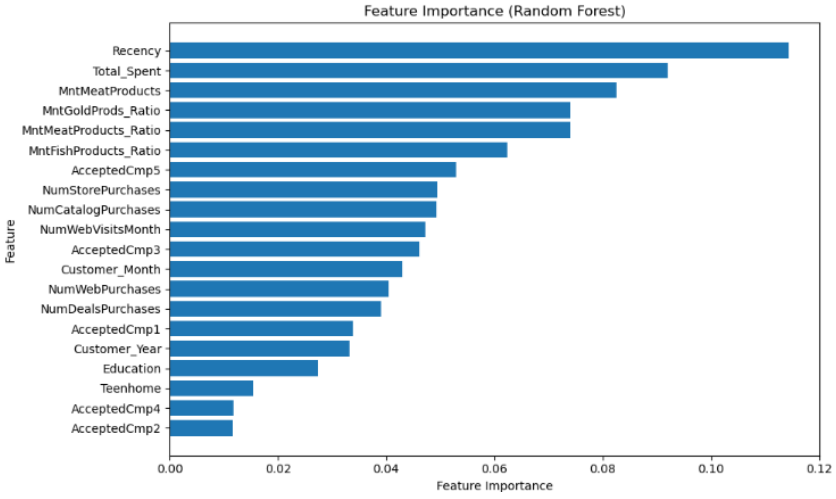
After fine-tuning, both **Logistic Regression** and **SVM** showed solid test set performance. However, the final decision was made considering **business needs**, **model interpretability**, and **training efficiency**:

- **Logistic Regression**
    - **Best Parameters**: {C: 1.25, solver: 'liblinear', max_iter: 200}
    - Maintained strong performance on both train and test sets
    - Scored highly in **accuracy**, **precision**, **recall**, **F1 Score**, and **AUC**
    - Its **simplicity**, **speed**, and **interpretability** made it an ideal final model
- **SVM**
    - Delivered competitive results after tuning
    - However, was **slower to train** and more difficult to interpret

Conclusion: Given the small performance gap and the added benefit of interpretability, **Logistic Regression** was chosen as the **final model**.

Feature Importance

Although **Logistic Regression** was chosen as the final model for deployment due to its simplicity and interpretability, we used a **Random Forest Classifier** to estimate **feature importance scores**. This approach helps identify which variables most significantly contribute to predicting whether a customer will respond to a campaign.



Business Takeaways:

This project successfully developed a **predictive model** to identify which customers are most likely to respond to a marketing campaign. The final model — **Logistic Regression** — was chosen for its:

- **Strong generalization** across unseen data
- **High interpretability**, making it ideal for marketing strategy decisions
- **Fast training and low computational cost**, suitable for deployment in real-time systems

Key Insights:

- **Recency** (how recently a customer engaged) is the strongest predictor of response — recent interactions matter.
- Customers with **higher spending levels**, especially on **meat and gold products**, are more responsive.
- Previous campaign interactions (e.g., **AcceptedCmp5**, **AcceptedCmp3**) are important behavioral signals.
- **Online and store purchase frequency** also influences response likelihood.

Business Impact:

- **Smarter Targeting**: The model allows marketers to focus on customers who are more likely to respond, improving ROI.
- **Reduced Costs**: By filtering out low-probability responders, campaigns become more cost-effective.
- **Actionable Profiles**: The feature insights help define clear customer segments for tailored messaging.

Thank You!