STRATEGIC CUSTOMER ANALYSIS: SEGMENTATION, CAMPAIGN RESPONSE, AND CHURN PREDICTION

USING MACHINE LEARNING TECHNIQUES

REPORT

Author:
RALITSA STOYCHEVA

23227443
MSC AI, BCU

Abstract

This report utilises machine learning to explore customer behaviour in the multi-channel retail domain. It applies K-Means clustering for customer segmentation and evaluates these segments with silhouette scores for targeted marketing strategies.

Supervised learning models are used to predict marketing campaign effectiveness and customer churn. Models like Random Forest and ensemble methods are examined, emphasising the balance between accuracy and business applicability. A quintile-based Linear Regression model is emphasised for reliable churn prediction, compared to a less effective continuous model.

The report concludes with recommendations for future research, including granular data analysis, residual analysis, and advanced regularisation techniques, showcasing the role of machine learning in enhancing customer engagement and guiding retail strategies.

Table of Contents

List of Tables

List of Figures

Introduction

Artificial Intelligence (AI) is increasingly crucial in business, offering revolutionary changes in operations. It enhances automation, productivity, and decision-making through advanced data analysis, leading to improved customer experiences and operational efficiency based on outputs from cognitive technologies (Zharovskikh, 2022). As of 2023, the AI market is valued between $136.55 billion (Thormundsson, 2023) to $207.9 billion, with projections of significant growth to $3,636 billion by 2033 (Grand View Research, 2022). In the UK, approximately 15% of businesses have adopted AI technologies, a trend expected to continue rising (GOV.UK, 2022).

The multi-channel retail environment, integral to the digital economy, combines various sales and communication channels for a unique shopping experience. Customers engage and shop through multiple platforms like physical stores, websites, and mobile apps (Crawley, 2021). Accelerated by the Covid-19 pandemic, this sector has seen rapid growth and transformation, influencing customer habits and innovating business models. The integrated retail approach has significantly altered shopping behaviours and transformed business-customer interactions (Briedis et al., 2023).

The aims and objectives of this report are centred around employing advanced analytical techniques to examine and interpret large-scale, multi-layered data from various retail channels using Python programming. Using both supervised and unsupervised machine learning techniques, this report aims to provide detailed insights into customer behaviour preferences and interactions across different retail channels. The goal of this report is to leverage these insights to provide the business with the ability to make informed, data-driven decisions, providing strategic vision and information that can shape their future in an ever-evolving multi-channel retail landscape.

Domain Description

This report focuses on the multi-channel retail domain, where businesses interact with customers through various channels, each offering unique customer experiences and engagement opportunities. This domain includes traditional retail stores, online marketplaces, social media platforms and mobile applications. It emphasises the importance of seamless shopping experience across all channels, catering to the diverse preferences and behaviours of customer (Iglesias-Pradas and Acquila-Natale, 2023).

Relevance of the Dataset to the Domain

The dataset pertinent to this study encapsulates a broad spectrum of customer engagement across multiple retail channels. It includes detailed information ranging from in-store interactions to online purchasing patterns, covering various aspects such as customer demographics, transaction histories, and responses to marketing efforts across different channels. This rich dataset is crucial in understanding the complex and evolving nature of customer behaviour in a multi-channel retail setting.

Importance of Machine Learning in the Domain

Machine learning is fundamental in the multi-channel retail environment, enabling analysis of complex datasets for actionable insights and enhancing customer experience and operational efficiency (Mitra et al., 2022). It also enables retailers to identify patterns in customer behaviour, facilitating personalised marketing strategies and improved customer experiences across different platforms. Techniques such as predictive analytics for customer behaviour, segmentation algorithms for personalised marketing, and churn prediction models are invaluable in this domain.

Problem Definition and Methodology

This section tackles three machine learning problems using algorithms tailored to each challenge. The selection was based on the learning type required and the unique aspects of each problem, ensuring an accurate application of machine learning to derive insightful solutions.

1. Customer Segmentation (Unsupervised Learning)

Objective: The overall goal of this unsupervised learning model is to effectively segment the customer base into meaningful groups based on their purchasing behaviours and demographic characteristics.

Target Variable:

- 'Affluent Enthusiasts' = High 'Income', 'MntWines', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds', and 'MntTotal'; Low 'NumDealsPurchases'
- 'Family-Focused Budget Shoppers' = High 'Kidhome', 'Teenhome', 'NumDealsPurchases'; Moderate to low 'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds'
- Digital Natives and Online Shoppers = High 'NumWebPurchases', 'NumWebVisitsMonth', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5'; Potentially younger age

Model Justification:

K-Means Clustering: To identify homogeneous groups within the customer base by finding the optimal partitioning of data (Melanie, 2023).

Silhouette Score: To evaluate the quality of clustering and ensure that each segment is distinct and internally cohesive (Scikit-learn, 2019).

Elbow Method: To determine the optimal number of clusters, thus balancing the granularity of segmentation with practical applicability (Saji, 2021).

Principal Component Analysis (PCA): To reduce the dimensionality of while retaining most of the significant variability in the data (Jaadi, 2019).

Standard Scaler: To normalise the feature set, ensuring each attribute contributes equally to the analysis (Scikit-Learn, 2019).

2. Campaign Effectiveness Prediction (Supervised Learning)

Objective: The primary objective is to predict whether a customer will accept an offer in a future marketing campaign. To achieve this, a large set of customer attributes and historical interaction data have been leveraged, serving as a foundation for identifying patterns and trends. Supervised learning has been chosen for this task as it excels in situations where the goal is to predict future occurrences based on past data (Jiang, Gradus, and Rosellini, 2020).

Target Variable: 'Response' is the target variable indicating whether a customer accepted an offer in past marketing campaigns. This binary variable provides a clear outcome for each customer where the goal is to predict a specific result based on input features.

Model Justification:

Logistic Regression: known for its simplicity and interpretability, served as a baseline, offering quick insights with reasonable accuracy (Jain, 2023).

Decision Trees and Random Forests: crucial for capturing non-linear relationships, with the latter providing enhanced generalisation through its ensemble approach (McClarren, 2021).

Gradient Boosting and AdaBoost: both boosting techniques, sequentially improved weak learners, focusing on misclassified instances in prior iterations, thereby enhancing the model's accuracy (Kumar, 2020).

K-Nearest Neighbours (KNN): leveraged similarity metrics to make predictions, offering a different perspective based on proximity in the feature space (IBM, 2023b).

Gaussian Naive Bayes: with its assumption of feature independence, provided a fast, baseline probabilistic approach (Oleszak, 2023).

Support Vector Machine (SVM): with its kernel trick, was adept at handling high-dimensional space (S, 2021).

XGBoost: was included for its efficiency and effectiveness in handling diverse datasets (XGBoost developers, 2022).

Voting Classifier: capitalised on their collective strengths, mitigating individual weaknesses, and offering a robust, well-rounded predictive tool (Medium, 2023).

This ensemble approach, especially after addressing class imbalance, proved highly effective in accurately predicting customer responses to future marketing campaigns.

3. Churn Prediction (Supervised Learning)

Objective: To forecast the number of days until customers are likely to end their association with the company (churn). By pinpointing those at risk of churning, pre-emptive measures to retain them can be implemented in the business.

Target Variable: The target variable in this study is 'Days Until Churn', which is predicted based on the RFM (Recency, Frequency, Monetary Value) model. 'Recency' is directly obtained from the dataset, 'Frequency' is engineered from the sum of NumWebPurchases, NumCatalogPurchases, NumDealsPurchases and NumStorePurchases, and 'Monetary' is derived from the 'MntTotal' feature.

Model Justification:

Linear Regression: To understand the relationship between the target variable ‘Days Until Churn’ and independent variables. Serves as foundation model (Mali, 2021).

Quantile-based Model: This approach divides the data into equal-sized segments based on rank-ordered values, making it particularly useful in handling skewed distributions and identifying patterns within specific segments of a dataset.

Continuous Model: Uses the actual numeric values of the data, aiming to leverage the full spectrum of information available in each variable.

Cross-Validation: Ensures the model’s robustness and generalisability across different data subsets.

Ridge Regression: Addresses overfitting by penalising large coefficients (L2 regularisation) (Scikit-learn, 2023).

Lasso Regression: Enhances model simplicity and interpretation by reducing some coefficients to zero (L1 regularisation) (Scikit-learn, 2023b).

Data Set Description

Data Source

The dataset analysed in this study was obtained from a publicly available source, Kaggle, a popular platform for data science and machine learning projects. This dataset is specifically tailored for marketing analytics, making it invaluable resource for understanding customer behaviour and preferences in a multi-channel retail context.

Initial Data Assessment

The dataset contains 2,205 instances (records) and 39 features (attributes). It is notable that the features in the dataset do not align with those described in the Kaggle description card (Fig. 1), suggesting either updates or variations in the dataset's version. The dataset's actual features are listed below (Table 1), along with the presumptive descriptions derived from the column names. These are presented in the exact same order as they appear in the CSV file.

Feature	Description	Comments
Income	Customer's yearly household income	
Kidhome	Number of small children in customer's household	
Teenhome	Number of teenagers in customer's household	
Recency	Number of days since the last purchase	
MntWines	Amount spent on wine products in the last 2 years	
MntFruits	Amount spent on fruits products in the last 2 years	
MntMeatProducts	Amount spent on meat products in the last 2 years	
MntFishProducts	Amount spent on fish products in the last 2 years	
MntSweetProducts	Amount spent on sweet products in the last 2 years	
MntGoldProds	Amount spent on gold products in the last 2 years	
NumDealsPurchases	Number of purchases made with discount	
NumWebPurchases	Number of purchases made through company's web site	
NumCatalogPurchases	Number of purchases made using catalogue	
NumStorePurchases	Number of purchases made directly in stores	
NumWebVisitsMonth	Number of visits to company's web site in the last month	

AcceptedCmp3	1 if customer accepted the offer in the 3rd campaign, 0 otherwise	
AcceptedCmp4	1 if customer accepted the offer in the 4th campaign, 0 otherwise	
AcceptedCmp5	1 if customer accepted the offer in the 5th campaign, 0 otherwise	
AcceptedCmp1	1 if customer accepted the offer in the 1st campaign, 0 otherwise	
AcceptedCmp2	1 if customer accepted the offer in the 2nd campaign, 0 otherwise	
Complain	1 if customer complained in the last 2 years	
Z_CostContact		This column was added from the list of actual columns.
Z_Revenue		This column was added from the list of actual columns.
Response	1 if customer accepted the offer in the last campaign, 0 otherwise	
Age	Age of customer	
Customer_Days	Number of days since registration as a customer	
marital_Divorced	1 if customer is divorced, 0 otherwise	
marital_Married	1 if customer is married, 0 otherwise	
marital_Single	1 if customer is single, 0 otherwise	
marital_Together	1 if customer is in relationship, 0 otherwise	
marital_Widow	1 if customer is a widow / widower, 0 otherwise	
education_2n Cycle	customer has secondary education	
education_Basic	customer has basic education	
education_Graduation	Customer has a Bachelor's degree	
education_Master	Customer has a Master's degree	
education_PhD	Customer has a PhD	

MntTotal	Total amount spent on all the products	
MntRegularProds	Amount spent on regular products in the last 2 years	
AcceptedCmpOverall	Overall number of accepted campaigns	
DtCustomer		There is no such column in the dataset.
Education		There is no such column in the dataset.
Marital		There is no such column in the dataset.

Table 1. Data Set Description

Characteristics of the Dataset

The dataset includes a wide range of attributes, from more specific metrics related to purchasing behaviour across multiple product categories to more fundamental demographic data such as age and marital status. Data on website visits, customer complaints, and customer interactions with marketing efforts are also included. This volume of information offers a thorough understanding of customer profiles and how they interact with the brand and its marketing initiatives. The dataset's combination of Booleans, float and integers allows for a multidimensional analysis method that considers both the categorical and quantitative aspects of customer behaviour, enabling the use of several machine learning techniques.

Justification for Dataset Choice

The dataset was chosen for its relevance to the selected domain of multi-channel retail and its stability for addressing the three identified machine learning problems – Campaign Effectiveness Prediction, Customer Segmentation, and Churn Prediction. The variety of features allows for an in-depth analysis of customer behaviour and the application of both supervised and unsupervised machine learning techniques.

Data Set Exploration

Preprocessing

Data preprocessing is a crucial stage in data analysis, focusing on cleaning, transforming, and streamlining data. It involves rectifying errors, inconsistencies, and inaccuracies to ensure data accuracy, completeness, and consistency. This process is vital for reliable decision-making, as it enhances data manageability and accuracy (Miller, 2019). This report outlines a preprocessing approach that ensures the integrity and quality of the data set for subsequent analyses.

Key reprocessing steps included the removal of redundant columns, ‘Z_Cost Contact’ and ‘Z_Revenue’, reducing the feature set to 37 (Fig. 2). This refinement was crucial in maintaining focus on the most relevant data. Furthermore, 184 duplicate entries, which can skew analysis, were identified, and removed (Fig. 3). This action refined the dataset from 2205 to 2021 instances, enhancing its authenticity.

The detection and exclusion of records with negative values in the 'MntRegularProds' column was another critical step. This measure addressed minor inconsistencies in the dataset, ensuring its consistency and validity (Fig. 4). Post these rigorous steps, the dataset was reduced to 2018 instances, each accurately representing a customer profile without redundancy or erroneous data (Fig. 5).

The entire preprocessing stage transformed the raw dataset into 'cleaned_data.csv', a polished resource ready for insightful analysis.

Exploratory Data Analysis (EDA)

EDA is integral in uncovering underlying patterns and relationships, which are essential for informed decision-making in marketing strategies (IBM, 2023a).

Correlation Matrix

The foundation of this analysis was the creation of a correlation matrix (Fig. 6) visualised using the “Mako” colour palette. This large-scale, detailed heatmap provided a comprehensive view of the interdependencies between features and highlighted potential collinearities and valuable relationships that are important for predictive modelling. The diagonal running from the top left to the bottom right consists of 1s, which represented the correlation of each variable with itself (perfect correlation). This matrix was used to identify potential relationships between variables that might warrant further analysis.

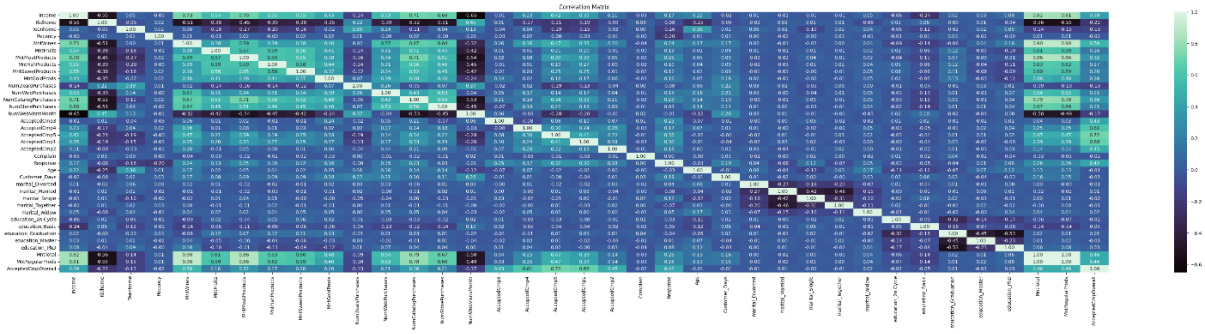


Fig. 6. Correlation Matrix

Outliers

Further, a series of boxplots for each feature played a pivotal role in the identification of outliers. In the context of marketing data, these outliers were deemed crucial as they represented actual customer behaviour. The strategic decision to retain these outliers ensured that all market segments, including atypical behaviours, were adequately considered in future campaign strategies (Fig. 7).

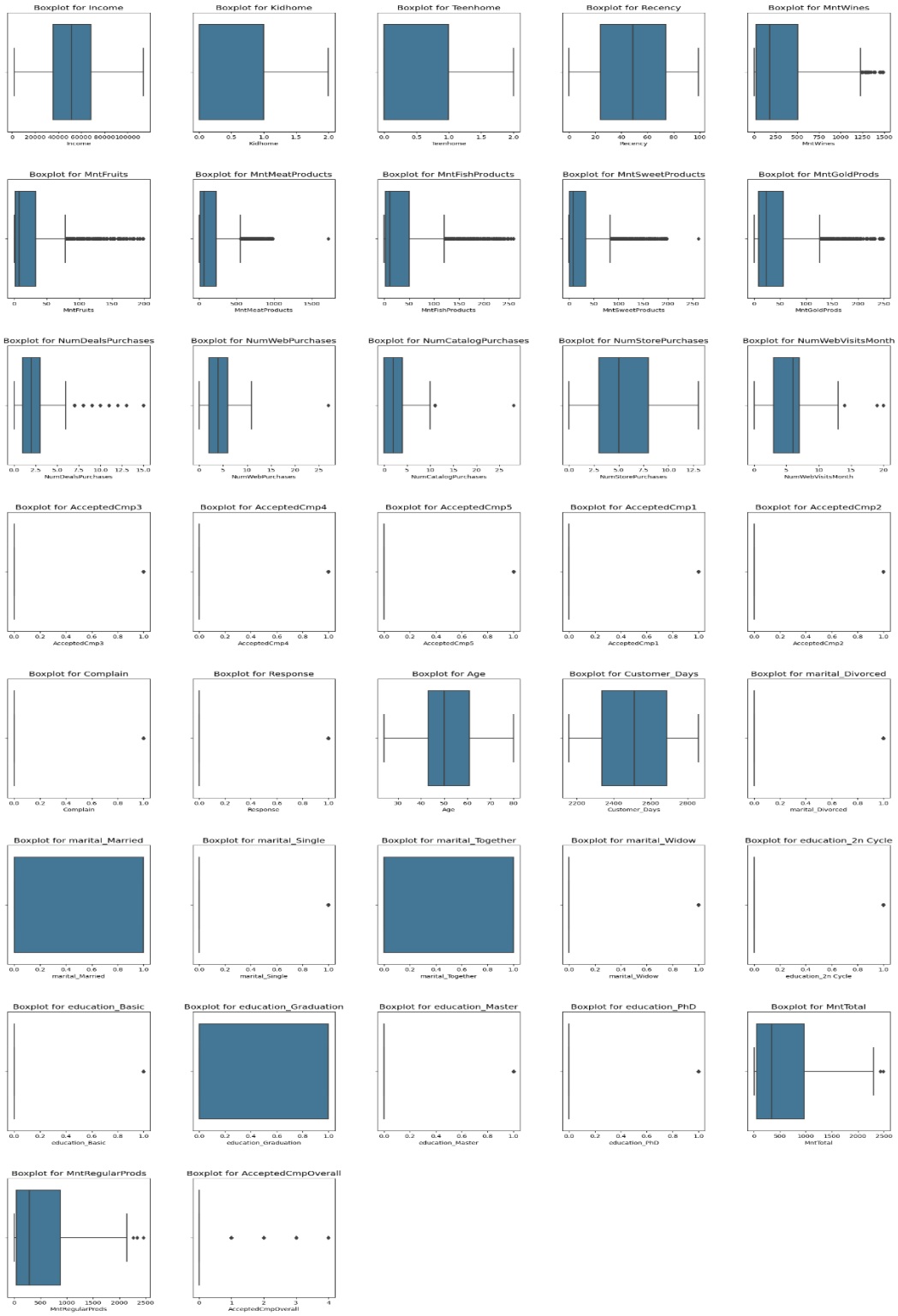


Fig. 7. Outliers

Density Plots for Key Features

Density plots for key features like 'Customer_Days', 'Age', 'Income', 'MntTotal', and 'Complain' offered deeper insights into the distribution of these variables. Using the same 'mako' colour palette, these plots helped in understanding the variability and skewness in the data, revealing the demographic and spending patterns of the customer base (Fig. 8).

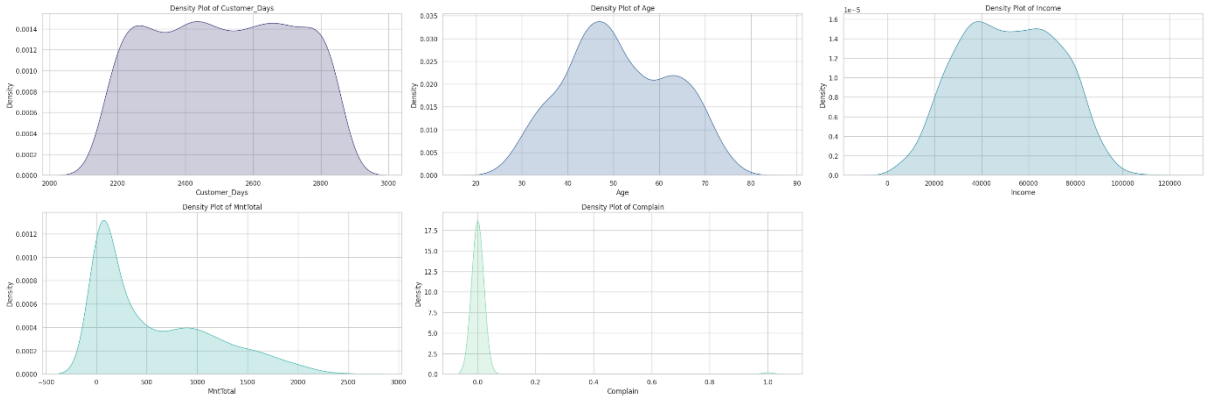


Fig. 8. Density Plots for Key Features

Grouped Bar Plot for Marketing Campaign Acceptance

Analysis of campaign acceptance revealed significant variations in success across different campaigns. Data were compiled into a new DataFrame to calculate the customer acceptance for each campaign, resulting in a grouped bar chart. The chart efficiently showed the varying impact of campaigns, highlighting Campaign 4 as particularly successful. While the overall campaign success was positive, individual response rates were lower than the cumulative campaign performance (Fig. 9).

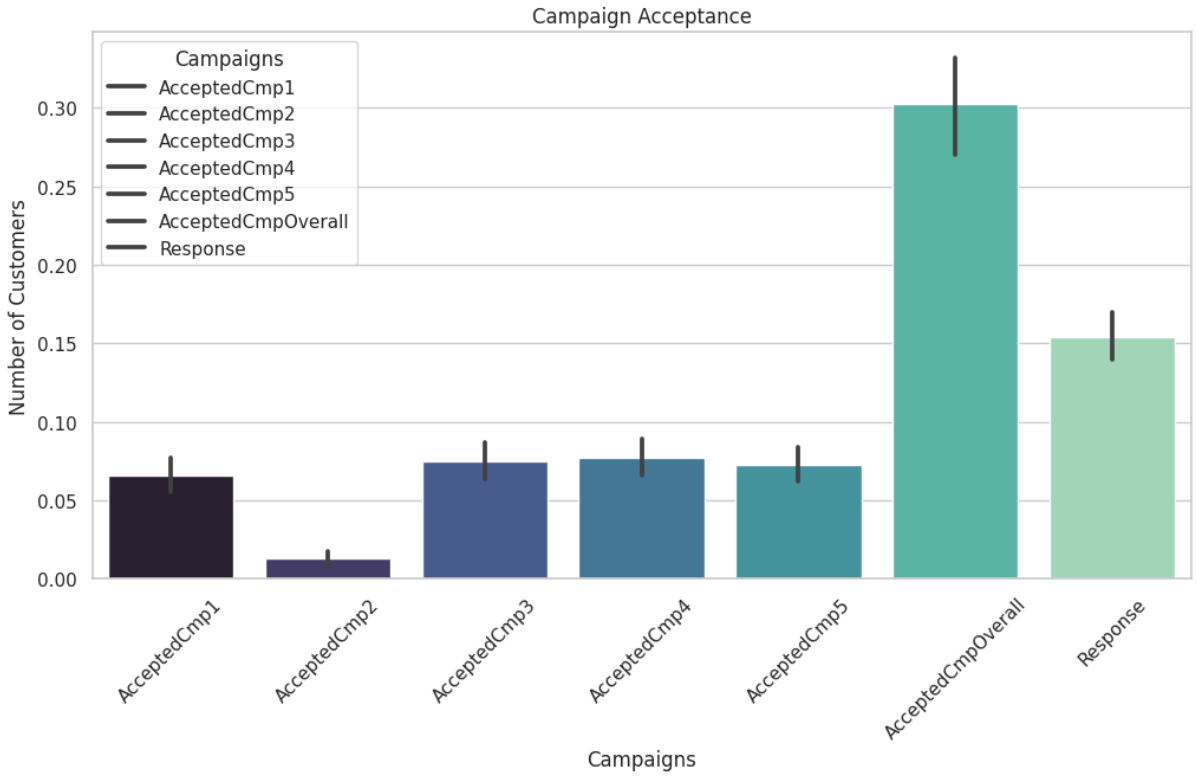


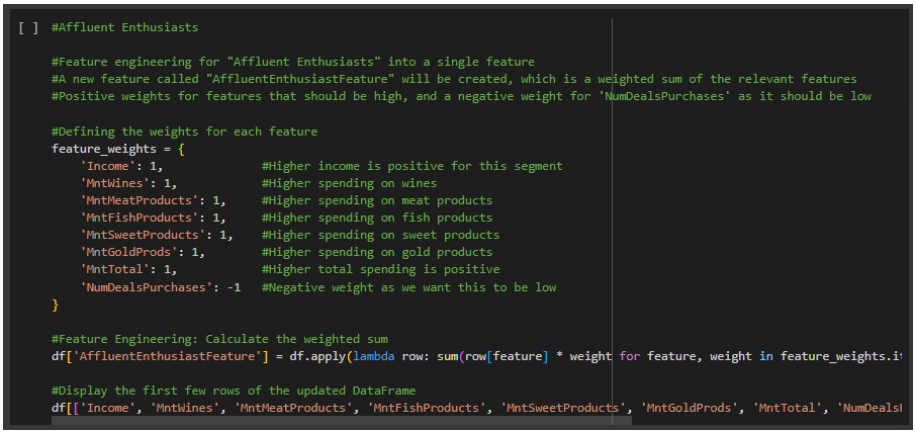
Fig. 9. Grouped Bar Plots for Marketing Campaign Acceptance

Experiments and Evaluations

Customer Segmentation (Unsupervised Learning)

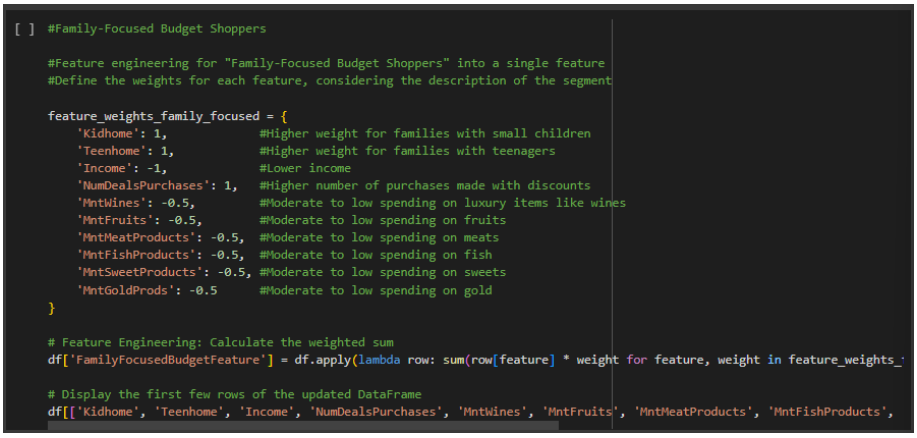
Objective and Feature Engineering

The objective of the unsupervised learning model detailed in this report was to segment the customer base into meaningful groups. These groups, or clusters, were based on purchasing behaviours and demographic characteristics. This segmentation aimed to identify distinct types of customers, such as 'Affluent Enthusiasts', 'Family-Focused Budget Shoppers', and 'Digital Natives and Online Shoppers'. Feature engineering (Fig. 10, 11, and 12) crafted new variables to reflect each segment's spending habits and preferences, such as weighting spending levels and online engagement while considering demographics like age.



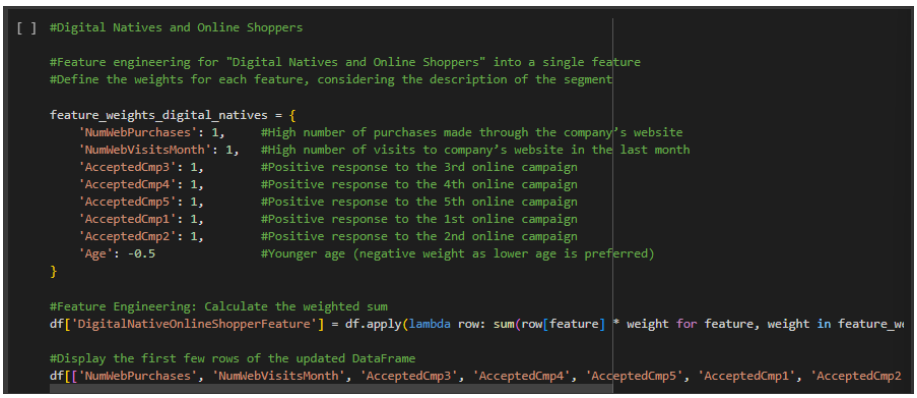
```
[ ] #Affluent Enthusiasts

    #Feature engineering for "Affluent Enthusiasts" into a single feature
    #A new feature called "AffluentEnthusiastFeature" will be created, which is a weighted sum of the relevant features
    #Positive weights for features that should be high, and a negative weight for 'NumDealsPurchases' as it should be low

    #Defining the weights for each feature
    feature_weights = {
        'Income': 1,              #Higher income is positive for this segment
        'MntWines': 1,            #Higher spending on wines
        'MntMeatProducts': 1,     #Higher spending on meat products
        'MntFishProducts': 1,     #Higher spending on fish products
        'MntSweetProducts': 1,    #Higher spending on sweet products
        'MntGoldProds': 1,        #Higher spending on gold products
        'MntTotal': 1,            #Higher total spending is positive
        'NumDealsPurchases': -1   #Negative weight as we want this to be low
    }

    #Feature Engineering: Calculate the weighted sum
    df['AffluentEnthusiastFeature'] = df.apply(lambda row: sum(row[feature] * weight for feature, weight in feature_weights.i

    #Display the first few rows of the updated DataFrame
    df[['Income', 'MntWines', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds', 'MntTotal', 'NumDealsI
```

Fig. 10. Feature Engineering – Affluent Enthusiasts



```
[ ] #Family-Focused Budget Shoppers

    #Feature engineering for "Family-Focused Budget Shoppers" into a single feature
    #Define the weights for each feature, considering the description of the segment

    feature_weights_family_focused = {
        'Kidhome': 1,             #Higher weight for families with small children
        'Teenhome': 1,            #Higher weight for families with teenagers
        'Income': -1,             #Lower income
        'NumDealsPurchases': 1,   #Higher number of purchases made with discounts
        'MntWines': -0.5,         #Moderate to low spending on luxury items like wines
        'MntFruits': -0.5,        #Moderate to low spending on fruits
        'MntMeatProducts': -0.5,  #Moderate to low spending on meats
        'MntFishProducts': -0.5,  #Moderate to low spending on fish
        'MntSweetProducts': -0.5, #Moderate to low spending on sweets
        'MntGoldProds': -0.5      #Moderate to low spending on gold
    }

    # Feature Engineering: Calculate the weighted sum
    df['FamilyFocusedBudgetFeature'] = df.apply(lambda row: sum(row[feature] * weight for feature, weight in feature_weights_

    # Display the first few rows of the updated DataFrame
    df[['Kidhome', 'Teenhome', 'Income', 'NumDealsPurchases', 'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts',
```

Fig. 11. Feature Engineering – Family-Focused Budget Shoppers



```
[ ] #Digital Natives and Online Shoppers

    #Feature engineering for "Digital Natives and Online Shoppers" into a single feature
    #Define the weights for each feature, considering the description of the segment

    feature_weights_digital_natives = {
        'NumWebPurchases': 1,     #High number of purchases made through the company's website
        'NumWebVisitsMonth': 1,   #High number of visits to company's website in the last month
        'AcceptedCmp3': 1,        #Positive response to the 3rd online campaign
        'AcceptedCmp4': 1,        #Positive response to the 4th online campaign
        'AcceptedCmp5': 1,        #Positive response to the 5th online campaign
        'AcceptedCmp1': 1,        #Positive response to the 1st online campaign
        'AcceptedCmp2': 1,        #Positive response to the 2nd online campaign
        'Age': -0.5               #Younger age (negative weight as lower age is preferred)
    }

    #Feature Engineering: Calculate the weighted sum
    df['DigitalNativeOnlineShopperFeature'] = df.apply(lambda row: sum(row[feature] * weight for feature, weight in feature_w

    #Display the first few rows of the updated DataFrame
    df[['NumWebPurchases', 'NumWebVisitsMonth', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'AcceptedCmp2
```

Fig. 12. Feature Engineering – Digital Natives and Online Shoppers

K-Means Clustering

The K-Means clustering algorithm was selected for its efficacy in identifying distinct groups within a dataset (Mannor et al., 2011). For this model, three clusters were predetermined to reflect the three customer segments of interest, despite the Elbow Method suggesting an optimal cluster number of two. This decision was informed by domain knowledge and the business context, which necessitated a tripartite segmentation.

Clusters were visualised through both 2D (Fig. 13) and 3D (Fig. 14) scatter plots, with a custom colour scale to distinguish between them. While 2D plots offer simplicity, they lack the capacity to represent the complexity of multi-dimensional data. The 3D plots, on the other hand, provided a more nuanced view of the customer segments in the context of the engineered features (Tian et al., 2021).

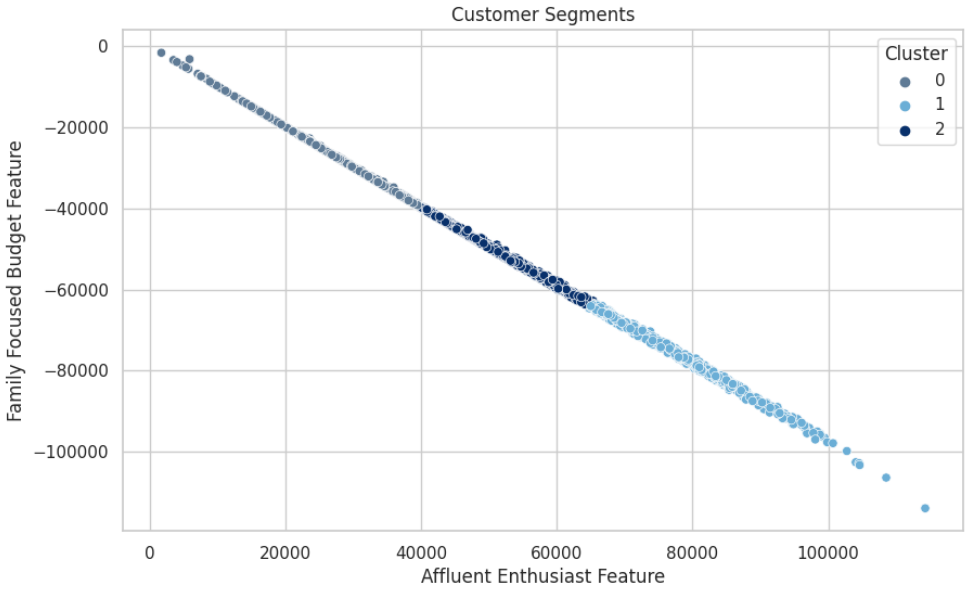


Fig. 13. 2D Clusters

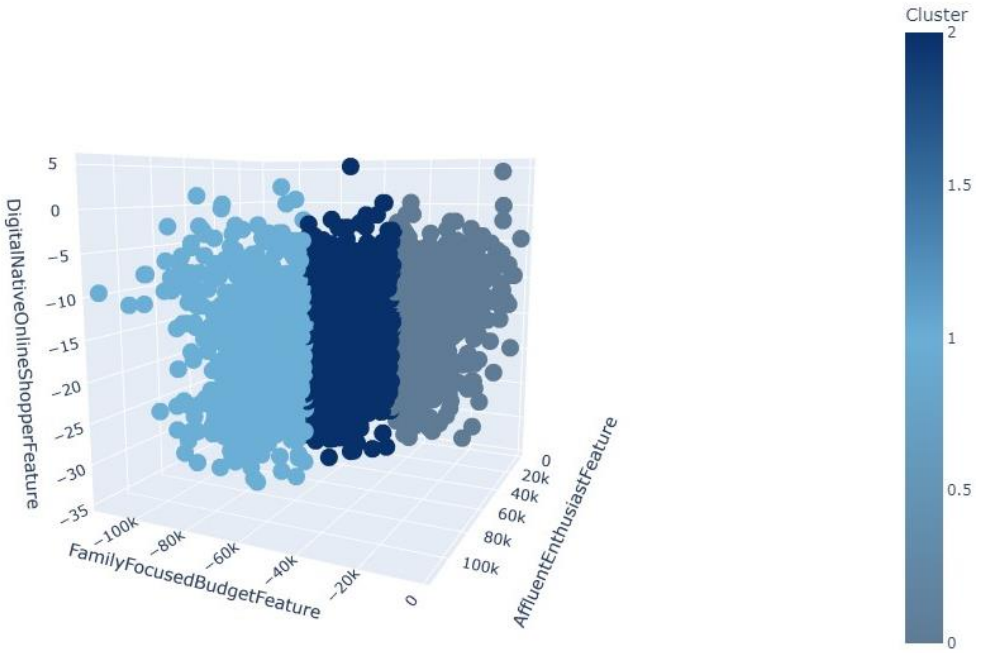


Fig. 14. 3D Clusters

Evaluation Metrics

The Sum of Squared Errors (SSE) was calculated across a range of cluster numbers to assess the quality of clustering, with the Elbow Method guiding the selection of an optimal cluster count (Fig. 15). However, the second derivative of the SSE indicated a preference for two clusters, which was ultimately overruled due to the business requirement of three distinct customer profiles.

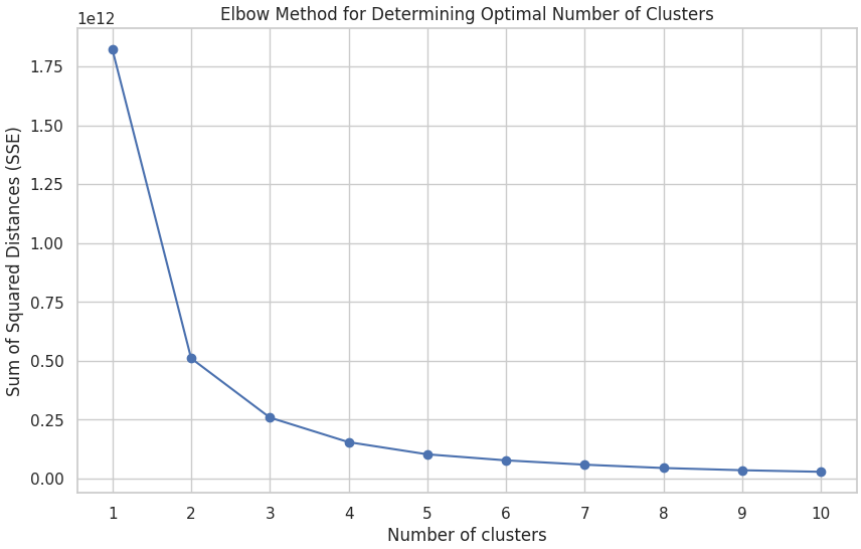


Fig. 15. The Elbow Method

The silhouette score, a metric used to evaluate the coherence of clusters (Yadav, 2023), initially suggests a moderately strong clustering structure with a score of 0.55 (Fig. 16). This score represented a balance between cohesion with clusters and separation between them.

Influence of PCA and Standard Scaler

An experiment to assess the influence of PCA and Standard Scaler on clustering revealed a decrease in the silhouette score from 0.55 to 0.39. This suggested that while PCA is a powerful tool for dimensionality reduction, its application in this context may have led to the loss of significant variance or distortion of cluster structures that are crucial for meaningful customer segmentation.

Campaign Effectiveness Prediction (Supervised Learning)

Reversion to the Original Data

In preparation for supervised learning, this analysis reverted to the original, cleaned DataFrame. The dimensions of the DataFrame were reaffirmed, ensuring that the data structure was intact for model training and evaluation.

Objective of Supervised Learning

The primary goal of this supervised learning approach was to predict customer acceptance of future marketing campaign offers. This predictive task is crucial for optimising marketing strategies and resource allocation.

Data Preparation and Processing

Missing values were imputed with feature means, and the dataset was divided into features and the 'Response' target variable, followed by a train-test split for model validation. Feature scaling standardised the data. No missing values remained, but correlations exceeding 0.8 indicated possible multicollinearity, potentially impacting models like logistic regression that require feature independence (Fig. 17).

Model Training and Evaluation

A suite of models, including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, KNeighbors, GaussianNB, SVC, and XGBoost, were trained and evaluated. The performance of each model was assessed based on accuracy, with Logistic Regression, Gradient Boosting, and AdaBoost performing best, achieving an accuracy of approximately 87.6% (Fig. 18).

Classification reports and confusion matrices were generated for each model to provide deeper insights into their performance. The reports revealed varying levels of precision, recall, and F1-score across models, indicating differences in their ability to predict customer responses accurately (Fig. 19).

```
Performance Report for LogisticRegression:
              precision    recall  f1-score   support

           0       0.91      0.95      0.93       503
           1       0.68      0.51      0.59       103

    accuracy                           0.88       606
   macro avg       0.79      0.73      0.76       606
weighted avg       0.87      0.88      0.87       606


Confusion Matrix for LogisticRegression:
[[478  25]
 [ 50  53]]

Performance Report for DecisionTreeClassifier:
              precision    recall  f1-score   support

           0       0.89      0.88      0.89       503
           1       0.44      0.47      0.45       103

    accuracy                           0.81       606
   macro avg       0.67      0.67      0.67       606
weighted avg       0.81      0.81      0.81       606


Confusion Matrix for DecisionTreeClassifier:
[[443  60]
 [ 55  48]]

Performance Report for RandomForestClassifier:
              precision    recall  f1-score   support

           0       0.87      0.97      0.92       503
           1       0.67      0.29      0.41       103

    accuracy                           0.85       606
   macro avg       0.77      0.63      0.66       606
weighted avg       0.84      0.85      0.83       606


Confusion Matrix for RandomForestClassifier:
[[488  15]
 [ 73  30]]

Performance Report for GaussianNB:
              precision    recall  f1-score   support

           0       0.91      0.83      0.87       503
           1       0.43      0.60      0.50       103

    accuracy                           0.80       606
   macro avg       0.67      0.72      0.69       606
weighted avg       0.83      0.80      0.81       606


Confusion Matrix for GaussianNB:
[[420  83]
 [ 41  62]]

Performance Report for SVC:
              precision    recall  f1-score   support

           0       0.88      0.97      0.92       503
           1       0.70      0.34      0.46       103

    accuracy                           0.86       606
   macro avg       0.79      0.65      0.69       606
weighted avg       0.85      0.86      0.84       606


Confusion Matrix for SVC:
[[488  15]
 [ 68  35]]
```

```
Performance Report for GradientBoostingClassifier:
              precision    recall  f1-score   support

           0       0.89      0.96      0.93       503
           1       0.72      0.45      0.55       103

    accuracy                           0.88       606
   macro avg       0.81      0.71      0.74       606
weighted avg       0.86      0.88      0.86       606


Confusion Matrix for GradientBoostingClassifier:
[[485  18]
 [ 57  46]]

Performance Report for AdaBoostClassifier:
              precision    recall  f1-score   support

           0       0.91      0.95      0.93       503
           1       0.68      0.51      0.59       103

    accuracy                           0.88       606
   macro avg       0.79      0.73      0.76       606
weighted avg       0.87      0.88      0.87       606


Confusion Matrix for AdaBoostClassifier:
[[478  25]
 [ 50  53]]

Performance Report for KNeighborsClassifier:
              precision    recall  f1-score   support

           0       0.87      0.97      0.92       503
           1       0.65      0.27      0.38       103

    accuracy                           0.85       606
   macro avg       0.76      0.62      0.65       606
weighted avg       0.83      0.85      0.83       606


Confusion Matrix for KNeighborsClassifier:
[[488  15]
 [ 75  28]]

Performance Report for XGBClassifier:
              precision    recall  f1-score   support

           0       0.90      0.94      0.92       503
           1       0.61      0.49      0.54       103

    accuracy                           0.86       606
   macro avg       0.75      0.71      0.73       606
weighted avg       0.85      0.86      0.85       606


Confusion Matrix for XGBClassifier:
[[471  32]
 [ 53  50]]
```

Fig. 19. Classification Reports and Confusion Matrixes of Models

Handling Imbalanced Data

Recognising the potential bias introduced by class imbalance (Fig. 20), RandomOverSampler (Fig. 21) was employed to balance the classes in the training data (Brownlee, 2020). This technique improved the minority class representation, potentially enhancing model performance for detecting positive responses.

Fig. 20. Handling Imbalanced Data

Fig. 21. RandomOverSampler

Subsequent model training on the oversampled data revealed changes in accuracy, with the Random Forest classifier showing the best performance at approximately 86.1% (Fig. 22). ROC and AUC were used to further analyse the results. However, the improvement in accuracy must be weighed against the increased likelihood of overfitting due to the synthetic oversampling of the minority class.

Performance Report for LogisticRegression on Oversampled Data:

	precision	recall	f1-score	support
0	0.95	0.80	0.87	503
1	0.45	0.82	0.58	103
accuracy			0.80	606
macro avg	0.70	0.81	0.73	606
weighted avg	0.87	0.80	0.82	606

Confusion Matrix for LogisticRegression on Oversampled Data:
[[402 101]
 [19 84]]

Performance Report for DecisionTreeClassifier on Oversampled Data:

	precision	recall	f1-score	support
0	0.87	0.88	0.88	503
1	0.39	0.37	0.38	103
accuracy			0.79	606
macro avg	0.63	0.62	0.63	606
weighted avg	0.79	0.79	0.79	606

Confusion Matrix for DecisionTreeClassifier on Oversampled Data:
[[443 60]
 [65 38]]

Performance Report for RandomForestClassifier on Oversampled Data:

	precision	recall	f1-score	support
0	0.88	0.96	0.92	503
1	0.66	0.39	0.49	103
accuracy			0.86	606
macro avg	0.77	0.67	0.70	606
weighted avg	0.85	0.86	0.85	606

Confusion Matrix for RandomForestClassifier on Oversampled Data:
[[482 21]
 [63 40]]

Performance Report for GaussianNB on Oversampled Data:

	precision	recall	f1-score	support
0	0.92	0.81	0.86	503
1	0.41	0.67	0.51	103
accuracy			0.78	606
macro avg	0.67	0.74	0.69	606
weighted avg	0.84	0.78	0.80	606

Confusion Matrix for GaussianNB on Oversampled Data:
[[405 98]
 [34 69]]

Performance Report for SVC on Oversampled Data:

	precision	recall	f1-score	support
0	0.92	0.84	0.88	503
1	0.46	0.66	0.54	103
accuracy			0.81	606
macro avg	0.69	0.75	0.71	606
weighted avg	0.85	0.81	0.82	606

Confusion Matrix for SVC on Oversampled Data:
[[424 79]
 [35 68]]

Performance Report for GradientBoostingClassifier on Oversampled Data:

	precision	recall	f1-score	support
0	0.94	0.84	0.89	503
1	0.49	0.73	0.59	103
accuracy			0.83	606
macro avg	0.71	0.79	0.74	606
weighted avg	0.86	0.83	0.84	606

Confusion Matrix for GradientBoostingClassifier on Oversampled Data:
[[425 78]
 [28 75]]

Performance Report for AdaBoostClassifier on Oversampled Data:

	precision	recall	f1-score	support
0	0.95	0.83	0.88	503
1	0.48	0.78	0.59	103
accuracy			0.82	606
macro avg	0.71	0.80	0.74	606
weighted avg	0.87	0.82	0.84	606

Confusion Matrix for AdaBoostClassifier on Oversampled Data:
[[417 86]
 [23 80]]

Performance Report for KNeighborsClassifier on Oversampled Data:

	precision	recall	f1-score	support
0	0.92	0.78	0.84	503
1	0.37	0.65	0.47	103
accuracy			0.75	606
macro avg	0.64	0.71	0.66	606
weighted avg	0.82	0.75	0.78	606

Confusion Matrix for KNeighborsClassifier on Oversampled Data:
[[390 113]
 [36 67]]

Performance Report for XGBClassifier on Oversampled Data:

	precision	recall	f1-score	support
0	0.91	0.91	0.91	503
1	0.57	0.54	0.55	103
accuracy			0.85	606
macro avg	0.74	0.73	0.73	606
weighted avg	0.85	0.85	0.85	606

Confusion Matrix for XGBClassifier on Oversampled Data:
[[460 43]
 [47 56]]

Fig. 22. Classification Reports and Confusion Matrixes of Models on Oversampled Data

Ensemble Learning

An ensemble approach (Fig. 23) was adopted by creating a Voting Classifier comprising all individual models. This ensemble method aimed to combine the predictions from each model to improve overall accuracy and robustness. The voting classifier's performance report and confusion matrix indicated a balanced trade-off between precision and recall, achieving an accuracy of 84%.

```
[37] #Reinitialise the models
     log_clf = LogisticRegression()
     dt_clf = DecisionTreeClassifier()
     rf_clf = RandomForestClassifier()
     gb_clf = GradientBoostingClassifier()
     ab_clf = AdaBoostClassifier()
     kn_clf = KNeighborsClassifier()
     gnb_clf = GaussianNB()
     svc_clf = SVC(probability=True)
     xgb_clf = XGBClassifier(use_label_encoder=False, eval_metric='logloss')

     #Train all the models on the oversampled data
     log_clf.fit(X_train_ros, y_train_ros)
     dt_clf.fit(X_train_ros, y_train_ros)
     rf_clf.fit(X_train_ros, y_train_ros)
     gb_clf.fit(X_train_ros, y_train_ros)
     ab_clf.fit(X_train_ros, y_train_ros)
     kn_clf.fit(X_train_ros, y_train_ros)
     gnb_clf.fit(X_train_ros, y_train_ros)
     svc_clf.fit(X_train_ros, y_train_ros)
     xgb_clf.fit(X_train_ros, y_train_ros)

     #Create a voting classifier
     voting_clf = VotingClassifier(
         estimators=[
             ('lr', log_clf),
             ('dt', dt_clf),
             ('rf', rf_clf),
             ('gb', gb_clf),
             ('ab', ab_clf),
             ('knn', kn_clf),
             ('gnb', gnb_clf),
             ('svc', svc_clf),
             ('xgb', xgb_clf)
         ],
         voting='hard'  # 'soft' for averaged probabilities or 'hard' for majority voting
     )

     #Train the voting classifier
     voting_clf.fit(X_train_ros, y_train_ros)

     #Make predictions and evaluate the voting classifier
     y_pred_voting = voting_clf.predict(X_test_scaled)

     #Print performance report
     print(classification_report(y_test, y_pred_voting))

     #Print confusion matrix
     print(confusion_matrix(y_test, y_pred_voting))

                   precision    recall  f1-score   support

                0       0.92      0.89      0.91       503
                1       0.54      0.61      0.57       103

         accuracy                           0.84       606
        macro avg       0.73      0.75      0.74       606
     weighted avg       0.85      0.84      0.85       606

     [[449  54]
      [ 40  63]]
```

Fig. 23. The Ensemble Learning Approach

This analysis emphasises the importance of aligning model evaluation with business objectives. While accuracy is important, the ultimate choice of model should consider the trade-offs between all performance metrics and their implications on campaign success and customer experience.

Churn Prediction (Supervised Learning)

Supervised Learning Objective and Feature Engineering

The objective of the supervised learning model was to predict the likelihood of customers discontinuing their relationship with the company - a critical insight for implementing proactive engagement strategies. For feature engineering, the RFM (Recency, Frequency, Monetary) model was utilised (Fig. 24). The 'DaysUntilChurn' feature was introduced with the logic that lower recency indicates a higher chance of churn (Fig. 25).

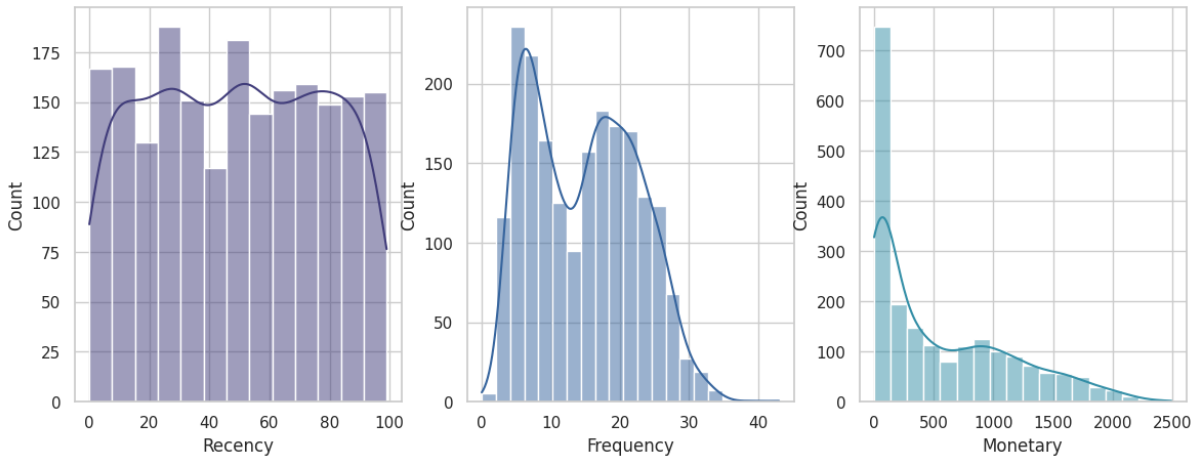


Fig. 24. Churn Prediction – RFM Feature Engineering

```
[39] #Feature Engineering
     df['Frequency'] = df['NumWebPurchases'] + df['NumCatalogPurchases'] + df['NumDealsPurchases'] + df['NumStorePurchases']
     df['Recency'] = df['Recency']
     df['Monetary'] = df['MntTotal']
     df['DaysUntilChurn'] = 100 - df['Recency']  #Simple logic
```

Fig. 25. Churn Prediction – RFM Feature Engineering and Model Logic

Model Development and Evaluation

After splitting the dataset into training and holdout sets, quintiles were calculated for each RFM metric. Two models were developed: one based on RFM quintiles and another using continuous RFM values. Both models underwent standard feature scaling to ensure model input standardisation.

Two metrics were used in evaluating the models:

- Mean Squared Error (MSE): A measure of prediction accuracy which assesses the average squared difference between the estimated values and the actual value (Allwright, 2022).
- R2 Score: A statistical measure of how close the data are to the fitted regression line (Turney, 2022).

The quintile-based model produced an MSE of 30.53 and an R2 score of 0.965 on the test set. For the continuous model, the MSE was astonishingly close to zero, and the R2 score was a perfect 1.0 on both test and holdout sets. These results were highly unusual, suggesting an overfitting problem in the continuous model, as perfect prediction is improbable in real-world scenarios (Fig. 26).

```
Test Set - Quintile-based Model - Mean Squared Error: 30.527407783683937
Test Set - Quintile-based Model - R2 Score: 0.9645813667942669
Test Set - Continuous Model - Mean Squared Error: 1.687149121331397e-28
Test Set - Continuous Model - R2 Score: 1.0
Holdout Set - Quintile-based Model - Mean Squared Error: 51.16011969557534
Holdout Set - Quintile-based Model - R2 Score: 0.935084899680118
Holdout Set - Continuous Model - Mean Squared Error: 1.8357808630014516e-28
Holdout Set - Continuous Model - R2 Score: 1.0
              Feature  Importance
1  FrequencyQuintile    0.153027
2   MonetaryQuintile    0.069858
0    RecencyQuintile  -20.193190
```

Fig. 26. Quantile-Based and Continuous Models Output

Visualisation of Predictive Model Accuracy on Holdout Data

The quintile-based model's accuracy was visualised against the actual data (Fig. 27). The plots showed varying degrees of predictive accuracy across different customer segments based on the RFM metrics. For instance, higher quintiles in the Monetary value plot correlated with lower predicted 'Days Until Churn,' indicating a lesser likelihood of churning for customers with higher spend.

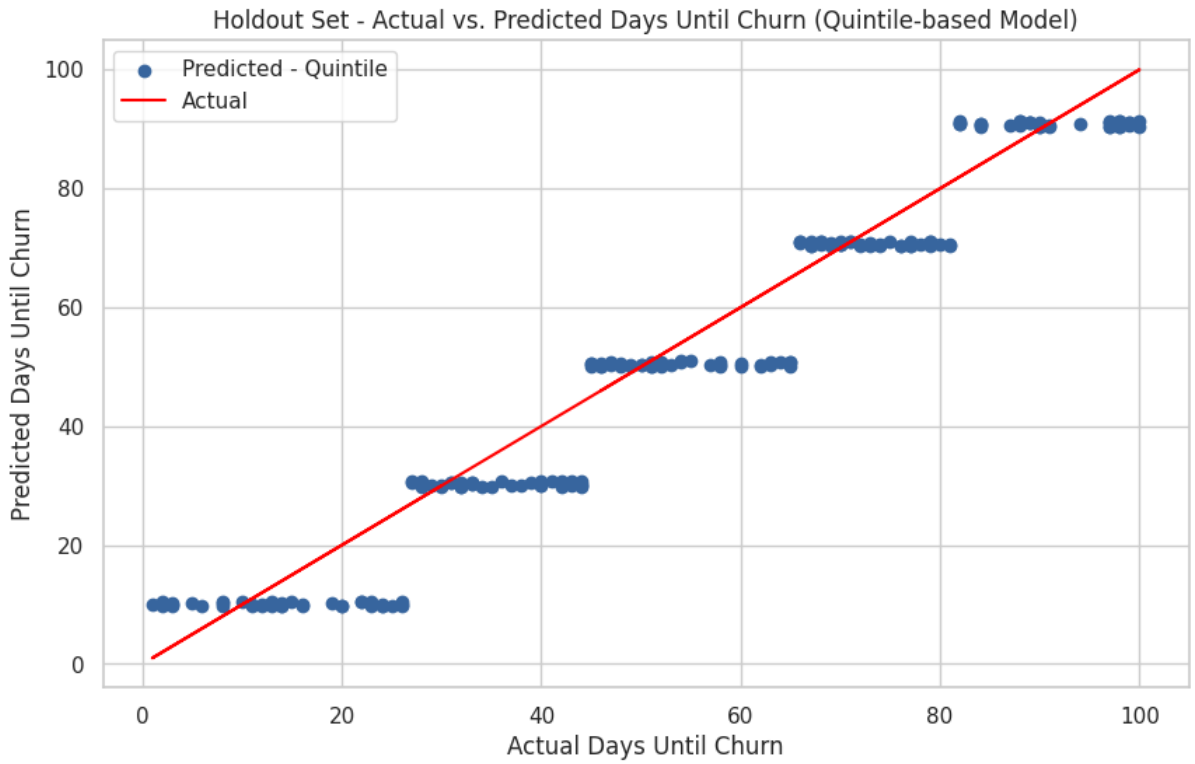


Fig. 27. Visualisation of Predictive Model Accuracy on Holdout Data

Segment-wise Analysis of Predicted Versus Actual Churn Days

The analysis revealed that different RFM segments had varying churn predictions, with the actual vs. predicted 'Days Until Churn' scatter plots indicating the quintile-based model's performance across segments. This segment-wise analysis provided actionable insights into the model's prediction accuracy, suggesting where improvements could be made (Fig. 28).

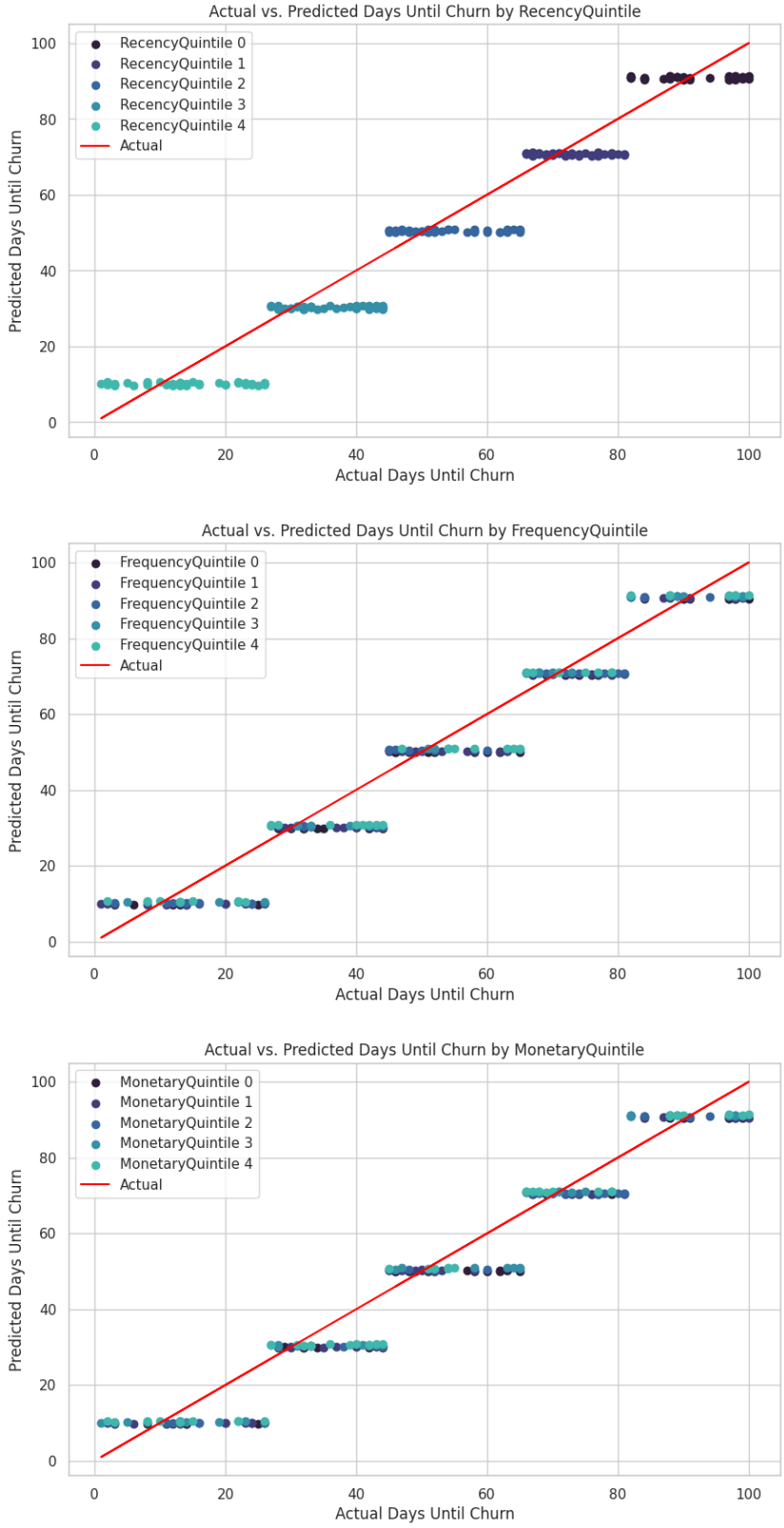


Fig. 28. Segment-wise Analysis of Predicted Versus Actual Churn Days

Cross-Validation of Models

The quintile-based model was cross-validated with an average R2 score of 0.960, confirming its strong performance. In contrast, the continuous model achieved a cross-validation R2 score of 1.0, which again suggested overfitting (Fig. 29).

```
Cross-validation R2 scores for the Quintile-based Model: [0.96298613 0.95187006 0.9657746  0.96204307 0.96844411 0.9524738
 0.95290713 0.96137618 0.96195308 0.96158246]
Average R2 score for the Quintile-based Model: 0.9601410640801269
Cross-validation R2 scores for the Continuous Model: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Average R2 score for the Continuous Model: 1.0
```

Fig. 29. Cross-Validation Models

Ridge and Lasso Regression Analysis

Both Ridge and Lasso regression techniques were applied to the models to combat overfitting. The quintile-based model maintained high R2 scores of 0.965 and 0.965, respectively, for Ridge and Lasso regression, which is consistent with a strong and reliable model. The continuous model saw marginal improvements in MSE and maintained a near-perfect R2 score, which still indicated potential overfitting (Fig. 30).

The quintile-based model emerged as more reliable, as its performance metrics were strong without being suspiciously perfect. This model was recommended for practical application, as its predictive power and generalisability are more aligned with real-world scenarios. The continuous model's perfect scores are a cause for concern and warrant further investigation into potential data leakage or the complexity of the model. Implementing regularisation techniques like Ridge or Lasso regression provided insights into improving the models' robustness and highlighted the quintile-based model's reliability.

Analysis and Results

This section consolidates the findings from various experiments across the unsupervised and supervised learning problems tackled in this study, synthesising them into coherent insights that align with the domain context and the initial objectives.

Unsupervised Learning – Customer Segmentation

The unsupervised learning model successfully identified customer segments, aiding in creating personalised marketing campaigns and resource allocation. The distinct spending patterns and demographic features of 'Affluent Enthusiasts' suggest targeted high-value product offerings, while the 'Family-Focused Budget Shoppers' could be engaged through discount-focused campaigns. Feature engineering was crucial for the effective use of K-Means clustering. The silhouette scores indicated moderate to strong cluster cohesion, substantiating the chosen segmentation despite alternatives presented by the Elbow Method. The 3D visualisations enhanced the understanding of the multi-dimensional nature of customer behaviours, solidifying the model's relevance to the domain.

Supervised Learning - Campaign Effectiveness Prediction

The supervised learning models addressed the key objective of predicting the success of marketing campaigns. The accuracy scores of Logistic Regression, Gradient Boosting, and AdaBoost, which hovered around 87.6%, were signifying a strong predictive capability. The ensemble approach provided a nuanced balance of precision and recall, guiding the deployment of marketing resources more efficiently. However, the challenge of class imbalance was notable. The application of RandomOverSampler improved representation and predictive performance, especially for the Random Forest classifier, which emerged as the standout model. It is important to note that while improved accuracy is desirable, it is the trade-off between various metrics that would ultimately guide the practical application of these models in real-world scenarios.

Supervised Learning - Churn Prediction

Churn prediction is critical for pre-emptive customer engagement. The quintile-based and continuous models demonstrated stark contrasts in performance, with the latter showing suspiciously perfect metrics indicative of overfitting. Consequently, the quintile-based model, with its realistic performance measures, stands out as the more applicable model for business use. The visualisation and segment-wise analysis of churn predictions highlighted the quintile-based model's capacity to differentiate between customer groups effectively. These insights are actionable, directing attention to segments at higher risk of churn. The model's ability to inform retention strategies is of high value, emphasising machine learning's role in enhancing customer retention in the multi-channel retail space.

Business Implications

Integrating these machine learning insights into the multi-channel retail domain could lead to significant advancements in customer experience and operational efficiency. The identification of customer segments can lead to more effective inventory management, tailored marketing strategies, and optimised channel engagement. The predictive models for campaign effectiveness and churn can inform customer relationship management systems, aiding in the design of personalised customer retention plans and optimising marketing budgets for maximum impact.

Conclusion

Summary of Key Findings

This report's analyses highlighted key insights into customer behaviours within the multi-channel retail domain. The application of K-Means clustering revealed distinct customer segments. The supervised learning models for campaign effectiveness and churn prediction highlighted Logistic Regression, Gradient Boosting, AdaBoost, and Linear Regression as robust performers. The quintile-based model was notably effective in churn prediction without overfitting.

Implications of the Study

The findings from this study underscore the significance of domain expertise in guiding machine learning approaches and the importance of evaluating models not just on accuracy but also on their alignment with business objectives. In marketing strategies, where missing out on potential acceptances could entail significant opportunity costs, models that balance precision and recall - especially those with higher recall for positive responses - may offer greater business value.

Limitations of the Study

Despite the insights gained, the study has limitations:

- Model Generalisability: The models may not perform equally well across different datasets or retail environments.
- Data Quality and Availability: The conclusions are as good as the data quality and may not account for uncollected or unobservable factors influencing customer behaviour.
- Complexity of Multi-Channel Data: The nature of multi-channel retail data, involving various platforms and customer touchpoints, poses a challenge for comprehensive and accurate analysis. The models might not fully capture the complexity of customer interactions across different channels.

Recommendations for Future Research

- Granular Data Analysis: Delving into more detailed customer data could reveal nuanced behavioural patterns that broad segmentation may miss.
- Residual Analysis for Insights: Examining the residuals of regression models could illuminate systematic prediction errors, informing more accurate and reliable predictive models.
- Additional Regularisation Techniques: Exploring regularisation methods such as Elastic Net may provide a nuanced regularisation approach, potentially enhancing model robustness and performance.

In conclusion, the strategic application of machine learning has the potential to transform multi-channel retail operations by providing deep customer insights and enabling data-driven decision-making. Future research should continue to refine these models, ensuring they remain relevant and valuable in a rapidly evolving retail landscape.

Reference List

Allwright, S. (2022). *What is a good MSE value? (simply explained)*. [online] Stephen Allwright. Available at: https://stephenallwright.com/good-mse-value/ [Accessed 28 Dec. 2023].

Briedis, H., Kronschnabl, A., Rodriguez, A. and Ungerman, K. (2023). *Adapting to the next normal in retail | McKinsey*. [online] www.mckinsey.com. Available at: https://mckinsey.com/industries/retail/our-insights/adapting-to-the-next-normal-in-retail-the-customer-experience-imperative [Accessed 11 Dec. 2023].

Brownlee, J. (2020). *Random Oversampling and Undersampling for Imbalanced Classification*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/ [Accessed 27 Dec. 2023].

Crawley, C. (2021). *Council Post: Omnichannel Versus Multichannel Marketing: Which Is Right For Your Brand?* [online] Forbes. Available at: https://www.forbes.com/sites/forbescommunicationscouncil/2021/05/18/omnichannel-versus-multichannel-marketing-which-is-right-for-your-brand/ [Accessed 11 Dec. 2023].

GOV.UK (2022). *AI activity in UK businesses: Executive Summary*. [online] GOV.UK. Available at: https://www.gov.uk/government/publications/ai-activity-in-uk-businesses/ai-activity-in-uk-businesses-executive-summary [Accessed 11 Dec. 2023].

Grand View Research (2022). *Artificial Intelligence Market Size, Share | AI Industry Report, 2025*. [online] Grand View Research. Available at: https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-market [Accessed 8 Jan. 2024].

IBM (2023a). *A Complete Guide on Exploratory Data Analysis*. [online] www.odinschool.com. Available at: https://www.odinschool.com/blog/a-complete-guide-on-exploratory-data-analysis [Accessed 15 Dec. 2023].

IBM (2023b). *What is the k-nearest neighbors algorithm? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/topics/knn [Accessed 28 Dec. 2023].

Iglesias-Pradas, S. and Acquila-Natale, E. (2023). The Future of E-Commerce: Overview and Prospects of Multichannel and Omnichannel Retail. *Journal of Theoretical and Applied Electronic Commerce Research*, [online] 18(1), pp.656–667. doi:https://doi.org/10.3390/jtaer18010033.

Jaadi, Z. (2019). *A Step by Step Explanation of Principal Component Analysis*. [online] Built In. Available at: https://builtin.com/data-science/step-step-explanation-principal-component-analysis [Accessed 28 Dec. 2023].

Jain, O. (2023). *Logistic Regression: A Comprehensive Introduction*. [online] Medium. Available at: https://osheenjain.medium.com/unlock-the-power-of-logistic-regression-a-comprehensive-introduction-e0e8ba98917d [Accessed 28 Dec. 2023].

Jiang, T., Gradus, J.L. and Rosellini, A.J. (2020). Supervised Machine Learning: A Brief Primer. *Behavior Therapy*, 51(5), pp.675–687. doi:https://doi.org/10.1016/j.beth.2020.05.002.

Kumar, R. (2020). *A Comparitive Study Between AdaBoost and Gradient Boost ML Algorithm*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/10/adaboost-and-gradient-boost-comparitive-study-between-2-popular-ensemble-model-techniques/ [Accessed 28 Dec. 2023].

Lemos, A.R. (2022). *Cross-Validation*. [online] Medium. Available at: https://towardsdatascience.com/cross-validation-705644663568 [Accessed 15 Dec. 2023].

Mali, K. (2021). *Linear Regression | Everything you need to Know about Linear Regression*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/ [Accessed 28 Dec. 2023].

Mannor, S., Jin, X., Han, J., Jin, X., Han, J., Jin, X., Han, J. and Zhang, X. (2011). K-Means Clustering. *Encyclopedia of Machine Learning*, pp.563–564. doi:https://doi.org/10.1007/978-0-387-30164-8_425.

McClarren, R.G. (2021). Decision Trees and Random Forests for Regression and Classification. *Machine Learning for Engineers*. doi:https://doi.org/10.1007/978-3-030-70388-2_3.

Medium (2023). *Improving Accuracy and Performance with Voting Classifiers in Ensemble Learning*. [online] Medium. Available at: https://blog.tdg.international/improving-accuracy-and-performance-with-voting-classifiers-in-ensemble-learning-ab5bd69e631f [Accessed 28 Dec. 2023].

Melanie (2023). *K-Means Clustering in Machine Learning: A Deep Dive*. [online] Data Science Courses | DataScientest. Available at: https://datascientest.com/en/k-means-clustering-in-machine-learning-a-deep-dive [Accessed 28 Dec. 2023].

Miller, R. (2019). *Data Preprocessing: what is it and why is important*. [online] CEOWORLD magazine. Available at: https://ceoworld.biz/2019/12/13/data-preprocessing-what-is-it-and-why-is-important/ [Accessed 15 Dec. 2023].

Mitra, A., Jain, A., Kishore, A. and Kumar, P. (2022). A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach. *Operations Research Forum*, [online] 3(4). doi:https://doi.org/10.1007/s43069-022-00166-4.

Oleszak, M. (2023). *Not-so-naive Bayes*. [online] Medium. Available at: https://towardsdatascience.com/not-so-naive-bayes-eb0936fa8b4a [Accessed 28 Dec. 2023].

S, P. (2021). *The A-Z guide to Support Vector Machine*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/#:~:text=Support%20Vector%20Machine%20(SVM)%20%E2%80%93%20(Interval%20block)%3A [Accessed 28 Dec. 2023].

Saji, B. (2021). *K Means Clustering | K Means Clustering Algorithm in Machine Learning*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/ [Accessed 28 Dec. 2023].

Scikit-Learn (2019). *sklearn.preprocessing.StandardScaler — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html [Accessed 28 Dec. 2023].

Scikit-learn (2019). *sklearn.metrics.silhouette_score — scikit-learn 0.21.3 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html [Accessed 28 Dec. 2023].

Scikit-learn (2023a). *Ridge coefficients as a function of the L2 Regularization*. [online] scikit-learn. Available at: https://scikit-learn.org/stable/auto_examples/linear_model/plot_ridge_coeffs.html#:~:text=To%20address%20overfitting%2C%20Ridge%20regularization [Accessed 28 Dec. 2023].

Scikit-learn (2023b). *sklearn.linear_model.Lasso*. [online] scikit-learn. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html [Accessed 28 Dec. 2023].

Thormundsson, B. (2023). *Artificial Intelligence Market Size 2030*. [online] Statista. Available at: https://www.statista.com/statistics/1365145/artificial-intelligence-market-size/ [Accessed 6 Jan. 2024].

Tian, Z., Zhai, X., van Steenpaal, G., Yu, L., Dimara, E., Espadoto, M. and Telea, A. (2021). Quantitative and Qualitative Comparison of 2D and 3D Projection Techniques for High-Dimensional Data. *Information*, 12(6), p.239. doi:https://doi.org/10.3390/info12060239.

Turney, S. (2022). *Coefficient of Determination (R^2) | Calculation & Interpretation*. [online] Scribbr. Available at: https://www.scribbr.com/statistics/coefficient-of-determination/ [Accessed 28 Dec. 2023].

Xgboost developers (2022). *XGBoost Documentation — xgboost 1.5.1 documentation*. [online] xgboost.readthedocs.io. Available at: https://xgboost.readthedocs.io/en/stable/ [Accessed 28 Dec. 2023].

Yadav, S. (2023). *Silhouette Coefficient Explained with a Practical Example: Assessing Cluster Fit"*. [online] Medium. Available at: https://medium.com/@Suraj_Yadav/silhouette-coefficient-explained-with-a-practical-example-assessing-cluster-fit-c0bb3fdef719 [Accessed 27 Dec. 2023].

Zharovskikh, A. (2022). *Importance of AI in Modern Corporate World*. [online] InData Labs. Available at: https://indatalabs.com/blog/importance-of-ai-in-corporate-world [Accessed 11 Dec. 2023].

Appendices

Link to Full Code and Comments in Colaboratory

https://colab.research.google.com/drive/1gv_-VaORPibntroro-x6UH6R1S3Hhrgi?usp=sharing

Link to Kaggle Dataset

https://www.kaggle.com/datasets/jackdaoud/marketing-data

Additional Figures

Feature	Description
AcceptedCmp1	1 if costumer accepted the offer in the 1st campaign, 0 otherwise
AcceptedCmp2	1 if costumer accepted the offer in the 2nd campaign, 0 otherwise
AcceptedCmp3	1 if costumer accepted the offer in the 3rd campaign, 0 otherwise
AcceptedCmp4	1 if costumer accepted the offer in the 4th campaign, 0 otherwise
AcceptedCmp5	1 if costumer accepted the offer in the 5th campaign, 0 otherwise
Response (target)	1 if costumer accepted the offer in the last campaign, 0 otherwise
Complain	1 if costumer complained in the last 2 years
DtCustomer	date of customer's enrollment with the company
Education	customer's level of education
Marital	customer's marital status
Kidhome	number of small children in customer's household
Teenhome	number of teenagers in customer's household
Income	customer's yearly household income
MntFishProducts	amount spent on fish products in the last 2 years
MntMeatProducts	amount spent on meat products in the last 2 years
MntFruits	amount spent on fruits in the last 2 years
MntSweetProducts	amount spent on sweet products in the last 2 years
MntWines	amount spent on wines in the last 2 years
MntGoldProds	amount spent on *gold* products in the last 2 years
NumDealsPurchases	number of purchases made with discount
NumCatalogPurchases	number of purchases made using catalogue
NumStorePurchases	number of purchases made directly in stores
NumWebPurchases	number of purchases made through company's web site
NumWebVisitsMonth	number of visits to company's web site in the last month
Recency	number of days since the last purchase

Fig. 1. Meta-Data Table

```
[ ]  print(df.shape) #Print DataFrame dimensions
     df.drop(columns=['Z_CostContact','Z_Revenue'],inplace=True) #Drop columns
     print(df.shape) #Print DataFrame dimensions

     (2205, 39)
     (2205, 37)
```

Fig. 2. Removal of Redundant Columns

```
[ ]  df.duplicated().sum() #Check for duplicates

     184

[ ]  print(df.shape) #Print DataFrame dimensions
     df=df.drop_duplicates()
     print(df.shape) #Print DataFrame dimensions

     (2205, 37)
     (2021, 37)
```

Fig. 3. Duplicates

```
[ ]  #Count negative values in each column
     negative_values_count = (df < 0).sum()

     #Display the count of negative values in each column
     print(negative_values_count)

     Income                0
     Kidhome               0
     Teenhome              0
     Recency               0
     MntWines              0
     MntFruits             0
     MntMeatProducts       0
     MntFishProducts       0
     MntSweetProducts      0
     MntGoldProds          0
     NumDealsPurchases     0
     NumWebPurchases       0
     NumCatalogPurchases   0
     NumStorePurchases     0
     NumWebVisitsMonth     0
     AcceptedCmp3          0
     AcceptedCmp4          0
     AcceptedCmp5          0
     AcceptedCmp1          0
     AcceptedCmp2          0
     Complain              0
     Response              0
     Age                   0
     Customer_Days         0
     marital_Divorced      0
     marital_Married       0
     marital_Single        0
     marital_Together      0
     marital_Widow         0
     education_2n Cycle    0
     education_Basic       0
     education_Graduation  0
     education_Master      0
     education_PhD         0
     MntTotal              0
     MntRegularProds       3
     AcceptedCmpOverall    0
     dtype: int64
```

Fig. 4. Negative Values

```
[ ]  print(df.shape) #Print DataFrame dimensions

     df = df[df['MntRegularProds'] >= 0]

     print(df.shape) #Print DataFrame dimensions

     (2021, 37)
     (2018, 37)
```

Fig. 5. Cleaned Data Frame Dimensions

```
[ ] #Cluster labels
    labels = df['Cluster']

    #Calculating the silhouette score
    silhouette_avg = silhouette_score(features, labels)
    print(f"The silhouette score of the clustering: {silhouette_avg:.2f}")

    The silhouette score of the clustering: 0.55
```

Fig. 16. Silhouette Score

```
[ ] #Handling missing values (if any)
    df.fillna(df.mean(), inplace=True)

    #Separating the target variable 'response' and features
    X = df.drop('Response', axis=1)
    y = df['Response']

    #Splitting the dataset 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)

    #Feature Scaling
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    #Checking for any remaining preprocessing needs
    missing_values = df.isnull().sum()
    highly_correlated_features = df.corr().abs().unstack().sort_values(kind="quicksort", ascending=False)
    high_corr = highly_correlated_features[(highly_correlated_features > 0.8) & (highly_correlated_features < 1)]

    #Outputting the results of preprocessing checks
    missing_values, high_corr
```

Fig. 17. Campaign Effectiveness Prediction – Data Preparation and Processing

```
[ ] #Initialise models
    logistic_regression = LogisticRegression()
    decision_tree = DecisionTreeClassifier()
    random_forest = RandomForestClassifier()
    gradient_boosting = GradientBoostingClassifier()
    ada_boost = AdaBoostClassifier()
    knn = KNeighborsClassifier()
    naive_bayes = GaussianNB()
    svm = SVC()
    xgboost = XGBClassifier(use_label_encoder=False, eval_metric='logloss')

    #List of models
    models = [logistic_regression, decision_tree, random_forest, gradient_boosting, ada_boost, knn, naive_bayes, svm, xgboost]

    #Dictionary to hold model names and their respective accuracies
    model_accuracies = {}

    #Training and evaluating each model
    for model in models:
        model_name = model.__class__.__name__
        model.fit(X_train_scaled, y_train)
        y_pred = model.predict(X_test_scaled)
        accuracy = accuracy_score(y_test, y_pred)
        model_accuracies[model_name] = accuracy

    model_accuracies

    {'LogisticRegression': 0.8762376237623762,
     'DecisionTreeClassifier': 0.8102310231023102,
     'RandomForestClassifier': 0.8547854785478548,
     'GradientBoostingClassifier': 0.8762376237623762,
     'AdaBoostClassifier': 0.8762376237623762,
     'KNeighborsClassifier': 0.8514851485148515,
     'GaussianNB': 0.7953795379537953,
     'SVC': 0.863036303630363,
     'XGBClassifier': 0.8597359735973598}
```

Fig. 18. Campaign Effectiveness Prediction – Model Training and Evaluation

```python
#Continuous Model - Ridge Regression
ridge_model_continuous = Ridge(alpha=1.0).fit(X_train_continuous, y_train_continuous)
ridge_predictions_continuous = ridge_model_continuous.predict(X_test_continuous)
ridge_mse_continuous = mean_squared_error(y_test_continuous, ridge_predictions_continuous)
ridge_r2_continuous = r2_score(y_test_continuous, ridge_predictions_continuous)
print("Continuous Model - Ridge Regression - MSE:", ridge_mse_continuous)
print("Continuous Model - Ridge Regression - R2 Score:", ridge_r2_continuous)

#Quintile-based Model - Ridge Regression
ridge_model_quintile = Ridge(alpha=1.0).fit(X_train_quintile, y_train_quintile)
ridge_predictions_quintile = ridge_model_quintile.predict(X_test_quintile)
ridge_mse_quintile = mean_squared_error(y_test_quintile, ridge_predictions_quintile)
ridge_r2_quintile = r2_score(y_test_quintile, ridge_predictions_quintile)
print("Quintile-based Model - Ridge Regression - MSE:", ridge_mse_quintile)
print("Quintile-based Model - Ridge Regression - R2 Score:", ridge_r2_quintile)


#Continuous Model - Lasso Regression
lasso_model_continuous = Lasso(alpha=0.01).fit(X_train_continuous, y_train_continuous)
lasso_predictions_continuous = lasso_model_continuous.predict(X_test_continuous)
lasso_mse_continuous = mean_squared_error(y_test_continuous, lasso_predictions_continuous)
lasso_r2_continuous = r2_score(y_test_continuous, lasso_predictions_continuous)
print("Continuous Model - Lasso Regression - MSE:", lasso_mse_continuous)
print("Continuous Model - Lasso Regression - R2 Score:", lasso_r2_continuous)

#Quintile-based Model - Lasso Regression
lasso_model_quintile = Lasso(alpha=0.01).fit(X_train_quintile, y_train_quintile)
lasso_predictions_quintile = lasso_model_quintile.predict(X_test_quintile)
lasso_mse_quintile = mean_squared_error(y_test_quintile, lasso_predictions_quintile)
lasso_r2_quintile = r2_score(y_test_quintile, lasso_predictions_quintile)
print("Quintile-based Model - Lasso Regression - MSE:", lasso_mse_quintile)
print("Quintile-based Model - Lasso Regression - R2 Score:", lasso_r2_quintile)
```

```
Continuous Model - Ridge Regression - MSE: 0.0005452053502322864
Continuous Model - Ridge Regression - R2 Score: 0.9999993674396314
Quintile-based Model - Ridge Regression - MSE: 30.524468223406373
Quintile-based Model - Ridge Regression - R2 Score: 0.9645847773428464
Continuous Model - Lasso Regression - MSE: 0.0001049619830266329
Continuous Model - Lasso Regression - R2 Score: 0.9999998782205812
Quintile-based Model - Lasso Regression - MSE: 30.52031352421558
Quintile-based Model - Lasso Regression - R2 Score: 0.9645895977248377
```

Fig. 30. Ridge and Lasso