Predictive Analysis from Consumer Profiles

Gabe Anoia, Harrison Hubbard, Levi Sessions

December 10, 2025

1 Introduction

In this report we aim to predict customer spending based on profiles providing past spending habits and demographic information. We examine customer spending across different product categories in relation to customer demographics such as gender, education, and income. Additionally, we use prior customer interactions such as response to marketing campaigns and store interaction to provide further analysis of customer spending patterns. These patterns help inform businesses to whom marketing should be targeted. The dataset used provides 2,240 row observations each consisting of 28 total columns, providing key data for making marketing decisions.

The dataset was collected by Dr. Omar Romero- Hernandez of U.C. Berkeley Haas School of Business for use by his students. Data was collected from registered customers of an undisclosed european retail chain with results from six email marketing campaigns. Demographic information was self-reported by the subjects, while the other variables were gathered from purchasing history data and survey responses from each email campaign. The data is mostly clean with 23 rows missing income values and four rows with ambiguous marital status categories.

Demographic information collected includes year of birth, gender, household income, education level, marital status, and number of children and number of teenagers in the house-

Variable Name	Description	Variable Type
Birth_Year	Customer's birth year	Numerical
Education	Customer's highest level of education obtained	Categorical
Marital_Status	Customer's marital status	Categorical
Income	Customer's yearly household income	Numerical
Kidhome	Number of children in customer's household	Numerical
Teenhome	Number of teenagers in customer's household	Numerical
Dt_Customer	Date of customer's enrollment with the business	Categorical
Recency	Number of days since customer's last purchase	Numerical
Complain	Whether or not customer has complained in the last two years	Categorical
MntWinesProducts	Amount spent on wine in last two years	Numerical
MntFruitsProducts	Amount spent on fruits in last two years	Numerical
MntMeatProducts Spending	Amount spent on meat in last two years	Numerical
MntFishProducts Spending	Amount spent on fish in last two years	Numerical
MntSweetProducts Spending	Amount spent on sweets in last two years	Numerical
MntGoldProds Spending	Amount spent on gold in last two years	Numerical
NumDealPurchases	Number of purchases made with a discount applied	Numerical
AcceptedCpn$N$ (1-5)	Whether or not customer accepted the offer in the Nth marketing campaign	Categorical
Response	Whether or not customer accepted the offer in the latest marketing campaign	Categorical
NumWebPurchases	Number of purchases made through the business's website	Numerical
NumCatalogPurchases	Number of purchases made using a catalogue	Numerical
NumStorePurchases	Number of purchases made directly in stores	Numerical
NumWebsiteVisitsMonth	Number of visits to business's website in the last month	Numerical

Table 1.1: Name and description of each variable in the consumer profile dataset.

hold. Education level includes primary education, undergraduate degree, master's degree, and doctoral degree (PhD). Marital status includes single, married, divorced, and widowed.

Spending data includes amount spent across six product categories: Wine, fruit, meat, fish, sweets, and gold. Purchase data was also collected keeping track of how many purchases were made with discounts applied and how many were made through website, catalog, the physical stores, separately. Further data was collected surrounding the six email marketing campaigns. For each, customers self-reported whether or not they made a purchase due to the respective campaign.

We use this data to answer the following questions: Can we accurately predict a customer's spending on one product category based via demographic information and spending on other categories? Can we find distinct groupings of customers based on spending across categories, means of purchasing, and recency of last purchase? We compare multiple machine learning models and statistical tools to find the best fit and answer these questions.

2 Exploratory Data Analysis

In this paper, we explore the relationship between different characteristics of a consumer. We hope to accurately predict consumer income via their shopping habits paired with their demographic. Utilising self supervised learning shows itself as a promising method to draw relationships between the many variables characterizing a customer.

We conduct analysis of both the numerical and categorical variables in this dataset. Our findings are documented in the following three sections.

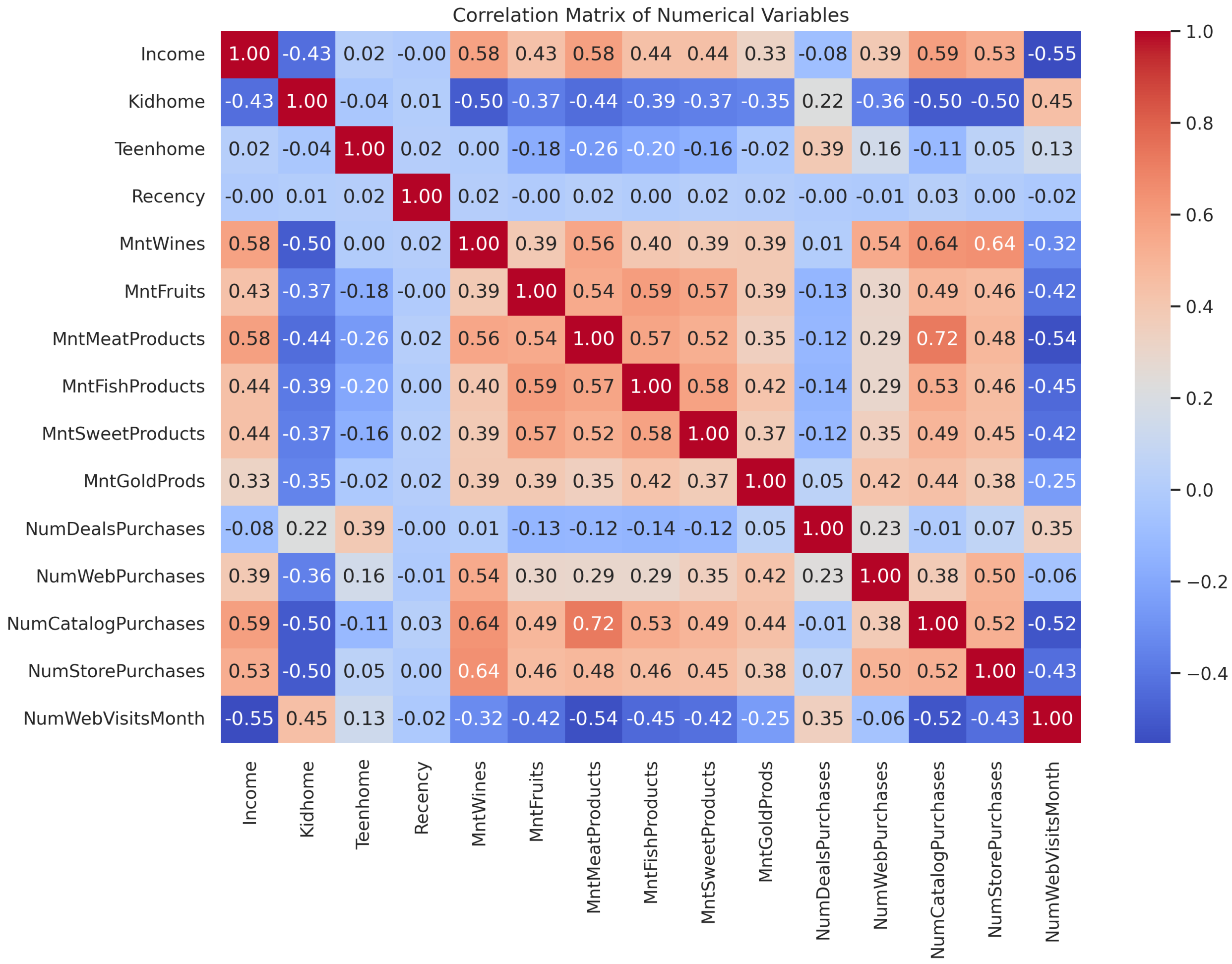
2.1 Numerical and Graphical Analysis of Numerical Variables

In this part of our analysis, we examine the dataset's numerical features to find distributions, relationships, and potential outliers. Through this analysis we can decide which variables may be most useful in our later predictive modeling tasks. We seek to gain deeper understanding of customers' behavior to guide our future experimentation.

Variable	Count	Mean	S.D.	Min	Q1	Median	Q3	Maximum
Income ($)	2,216	52,247.25	25,173.08	1,730.00	35,303.00	51,381.50	68,522.00	666,666.00
Kidhome	2,240	0.44	0.54	0	0	0	1	2
Teenhome	2,240	0.51	0.54	0	0	0	1	2
Recency	2,240	49.11	28.96	0	24	49	74	99
MntWines ($)	2,240	303.94	336.60	0.00	23.75	173.50	504.25	1,493.00
MntFruits ($)	2,240	26.30	39.77	0.00	1.00	8.00	33.00	199.00
MntMeatProducts ($)	2,240	166.95	22.5.72	0.00	16.00	67.00	232.00	1,725.00
MntFishProducts ($)	2,240	37.53	54.63	0.00	3.00	12.00	50.00	259.00
MntSweetProducts ($)	2,240	27.06	41.28	0.00	1.00	8.00	33.00	263.00
MntGoldProds ($)	2,240	44.02	52.17	0.00	9.00	24.00	56.00	362.00
NumDealsPurchases	2,240	2.34	1.93	0	1	2	3	15
NumWebPurchases	2,240	4.08	2.78	0	2	4	6	27
NumCatalogPurchases	2,240	2.66	2.92	0	0	2	4	28
NumStorePurchases	2,240	5.79	3.25	0	3	5	8	13
NumWebVisitsMonth	2,240	5.32	2.42	0	3	6	7	20

Table 2.1: Numerical summary of each numerical variable.

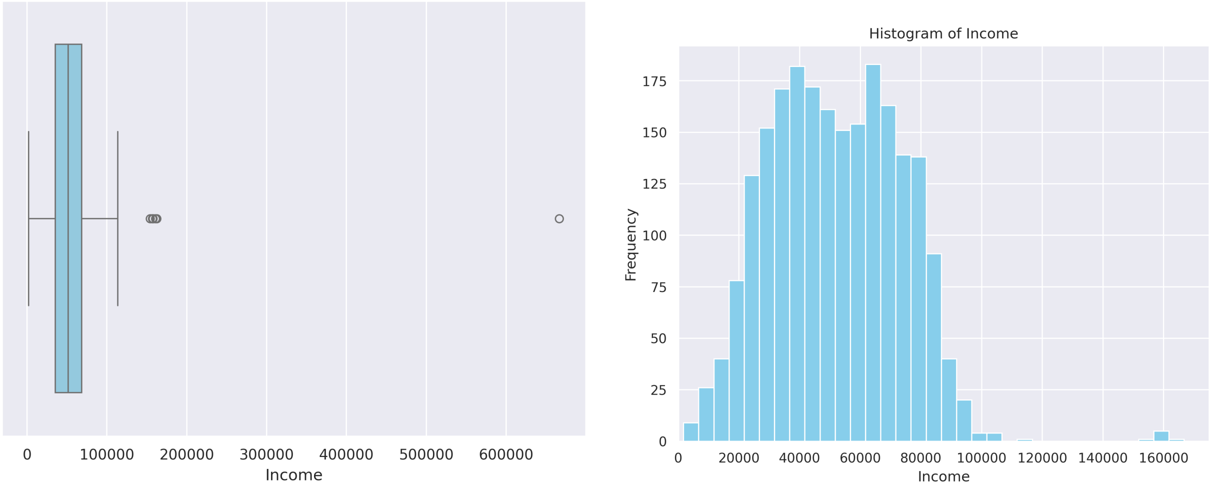
The numerical features of this dataset show a high degree of variation. Income, for example, has a median of $51,381.50 while the maximum value is $666,666, skewing the values higher. The number of children and teens in the home are notably small, with most customers reporting zero or one child or teenager at home. Categorical spending also shows skewed patterns with wine and cheese being disproportionately larger than fruit, fish, sweets, and gold. Means of purchasing show moderate values with in-store purchasing having the highest median response, although catalog and web purchases both have higher maximum values than in-store.



	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth
Income	1.00	-0.43	0.02	-0.00	0.58	0.43	0.58	0.44	0.44	0.33	-0.08	0.39	0.59	0.53	-0.55
Kidhome	-0.43	1.00	-0.04	0.01	-0.50	-0.37	-0.44	-0.39	-0.37	-0.35	0.22	-0.36	-0.50	-0.50	0.45
Teenhome	0.02	-0.04	1.00	0.02	0.00	-0.18	-0.26	-0.20	-0.16	-0.02	0.39	0.16	-0.11	0.05	0.13
Recency	-0.00	0.01	0.02	1.00	0.02	-0.00	0.02	0.00	0.02	0.02	-0.00	-0.01	0.03	0.00	-0.02
MntWines	0.58	-0.50	0.00	0.02	1.00	0.39	0.56	0.40	0.39	0.39	0.01	0.54	0.64	0.64	-0.32
MntFruits	0.43	-0.37	-0.18	-0.00	0.39	1.00	0.54	0.59	0.57	0.39	-0.13	0.30	0.49	0.46	-0.42
MntMeatProducts	0.58	-0.44	-0.26	0.02	0.56	0.54	1.00	0.57	0.52	0.35	-0.12	0.29	0.72	0.48	-0.54
MntFishProducts	0.44	-0.39	-0.20	0.00	0.40	0.59	0.57	1.00	0.58	0.42	-0.14	0.29	0.53	0.46	-0.45
MntSweetProducts	0.44	-0.37	-0.16	0.02	0.39	0.57	0.52	0.58	1.00	0.37	-0.12	0.35	0.49	0.45	-0.42
MntGoldProds	0.33	-0.35	-0.02	0.02	0.39	0.39	0.35	0.42	0.37	1.00	0.05	0.42	0.44	0.38	-0.25
NumDealsPurchases	-0.08	0.22	0.39	-0.00	0.01	-0.13	-0.12	-0.14	-0.12	0.05	1.00	0.23	-0.01	0.07	0.35
NumWebPurchases	0.39	-0.36	0.16	-0.01	0.54	0.30	0.29	0.29	0.35	0.42	0.23	1.00	0.38	0.50	-0.06
NumCatalogPurchases	0.59	-0.50	-0.11	0.03	0.64	0.49	0.72	0.53	0.49	0.44	-0.01	0.38	1.00	0.52	-0.52
NumStorePurchases	0.53	-0.50	0.05	0.00	0.64	0.46	0.48	0.46	0.45	0.38	0.07	0.50	0.52	1.00	-0.43
NumWebVisitsMonth	-0.55	0.45	0.13	-0.02	-0.32	-0.42	-0.54	-0.45	-0.42	-0.25	0.35	-0.06	-0.52	-0.43	1.00

Figure 2.1: Correlation heatmap of numerical variables.

This correlation heatmap provides valuable insights into how customer demographics, income, and purchasing behaviors are related across different channels and product categories. For example, customer income shows a strong positive correlation with catalog purchases (0.59), meat products (0.58), and wine (0.58), indicating that higher-income customers spend more on premium items and prefer catalog/store channels. At the same time, income is negatively correlated with website visits per month (-0.55), suggesting wealthier customers browse less online. Families with kids at home display negative correlations with product spending (e.g., wine -0.50) and income (-0.43), but a positive relationship with web visits (0.45). Households with teens lean toward deal driven behavior, shown by a positive correlation with deals purchases (0.39). Overall, the strong inter correlations between product categories (e.g., meat and fish 0.59) reveal that heavy spenders often purchase across multiple product lines, highlighting distinct customer segments for tailored marketing strategies.



(a) Boxplot of income feature

(b) Histogram of customer incomes

Figure 2.2: Income: Boxplot and histogram shown side by side.

The histogram and boxplot of income provide a clear view of the distribution and variability of customer earnings. The histogram shows that most customers fall within the $20,000–$80,000 range, with peaks around $40,000–$60,000, representing the bulk of the customer base. On the other hand, the boxplot highlights the presence of several high-income outliers, including extreme cases above $200,000 and even one beyond $600,000, which skew the income distribution. These outliers may represent a very small but valuable segment of premium customers.

When tied back to the correlation heatmap, income was shown to be positively correlated with spending on wines (0.58), meat products (0.58), and catalog purchases (0.59), confirming that higher income customers drive much of the premium product sales. However, income also had a negative correlation with web visits (-0.55), reinforcing the idea that wealthier customers prefer catalog and store channels over frequent online browsing. Overall, while the majority of customers are middle income, the higher income minority appears to play an outsized role in driving revenue for luxury and catalog based purchases.

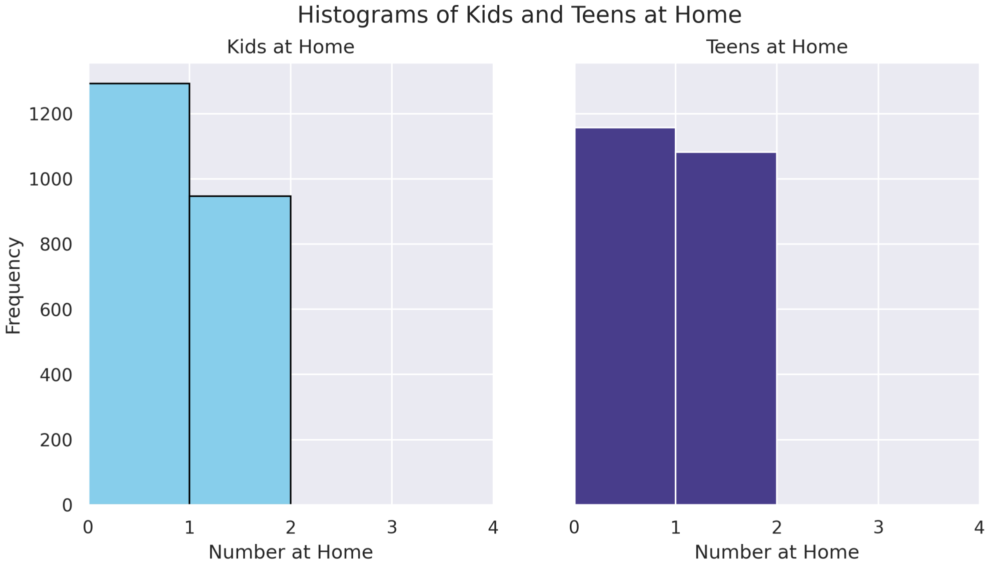


Figure 2.3: Histogram of Customers with kids and teen feature

This set of histograms shows the distribution of kids and teens at home and connects directly to the correlations observed in the heatmap. The Kids at Home histogram indicates that most households have either 0 or 1 child, with fewer families having 2 or more. Similarly, the Teens at Home histogram shows that the majority of households have 0 or 1 teen, with very few families having 2 or more. This aligns with the correlation matrix, where having kids at home showed a negative relationship with income (-0.43) and product spending (e.g., wine -0.50, meat -0.44). Since families with kids are more common, these negative correlations suggest that a large portion of households may be constrained in income and discretionary spending.

On the other hand, the weaker correlations for teens (e.g., slightly positive with deals purchases at 0.39) reflect what we see here: while many households have teens, their presence does not strongly influence income or overall product spending but does push families toward deal-driven behaviors. Taken together, the histograms confirm that families with kids and teens make up a substantial part of the customer base, and as highlighted in the heatmap, their behaviors differ significantly from higher-income, child-free households who dominate premium spending.

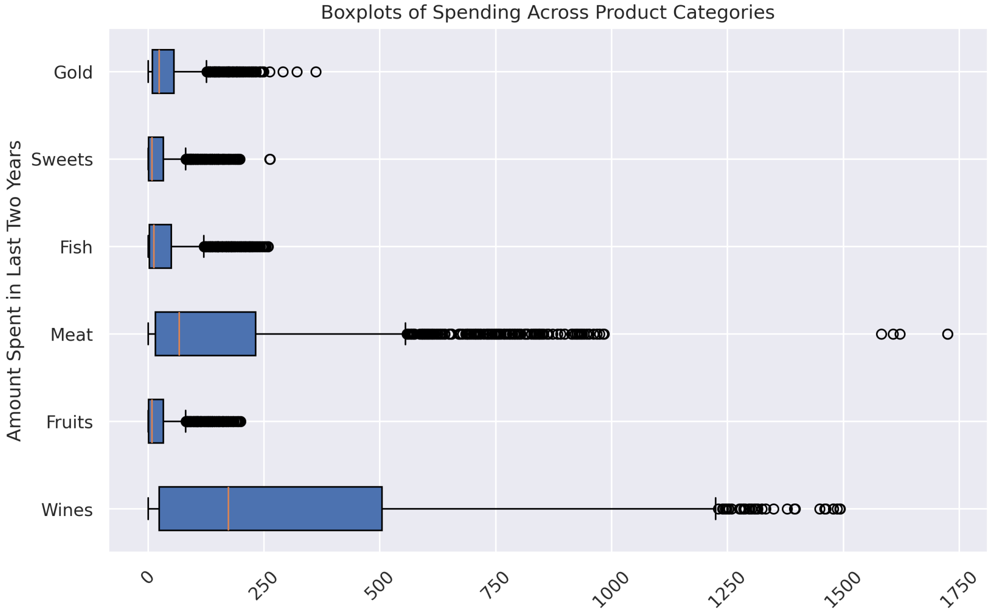


Figure 2.4: Boxplot of spending across various categories

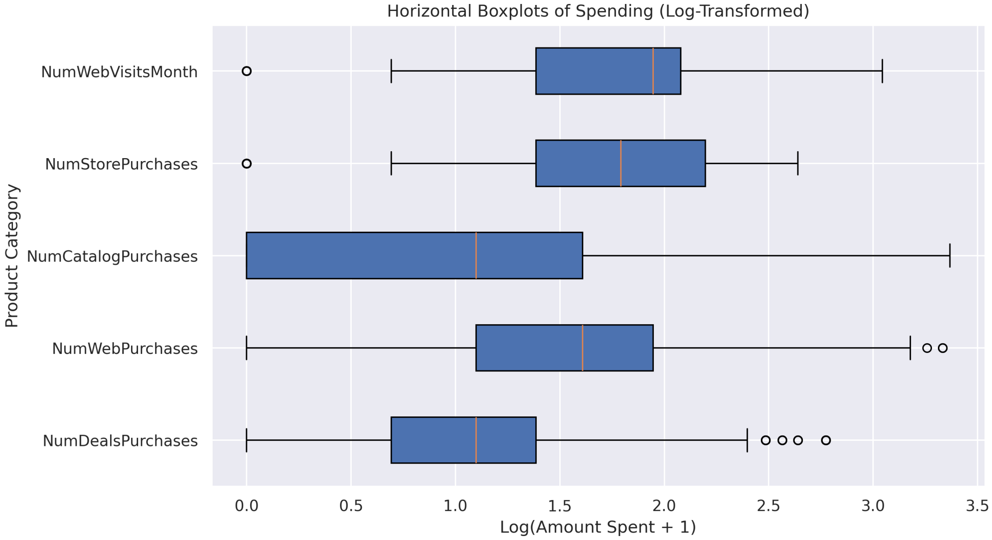


Figure 2.5: Boxplot of spending across various categories (log transformation)

The regular boxplots of product spending show that wine and meat dominate customer expenditures, with wine having the widest spread and the largest number of extreme outliers (some exceeding 1,700 units). Meat also has a large range and many high outliers, while categories like gold, sweets, fruits, and fish show smaller medians and narrower spreads. This aligns with the correlation matrix, where income is most strongly tied to wine (0.58) and meat (0.58), confirming that higher-income customers drive much of the high end spending.

The log transformed boxplots help address the issue of skewness. In the raw plots, extreme outliers stretch the scale, making it difficult to see differences among the majority of customers who spend at moderate levels. The log transformation compresses these extreme values, making distributions more balanced and comparable across categories. On the log scale, wine and meat still stand out as the highest spending categories, but now it's easier to see variation within lower spending groups like sweets and fish. This transformation is crucial for highlighting underlying spending patterns without having them obscured by a small number of very heavy spenders.

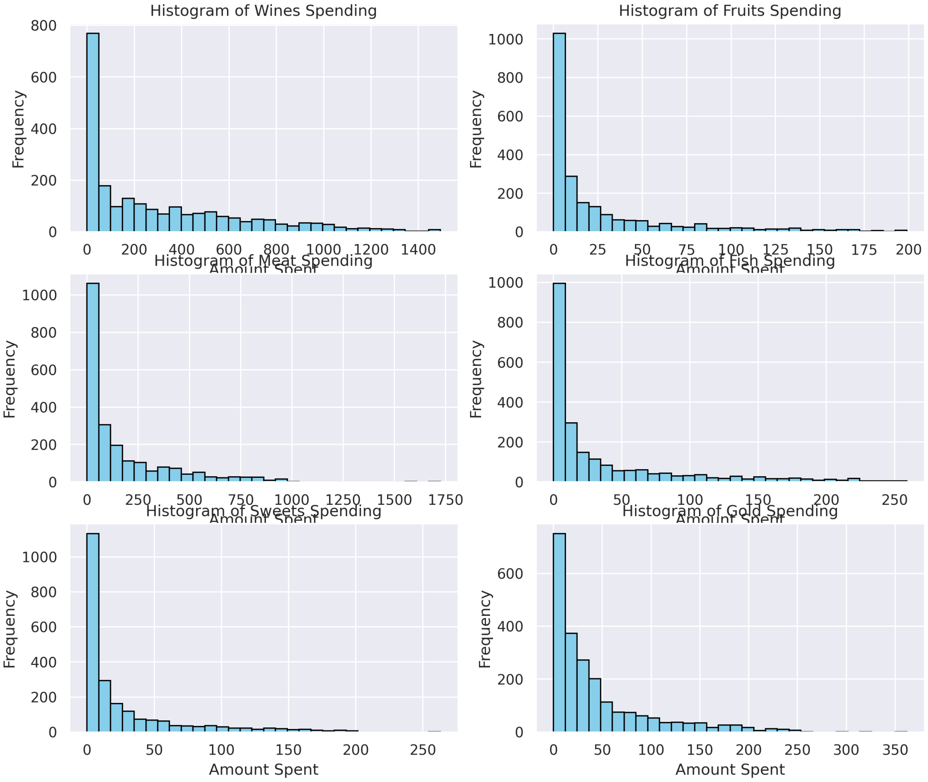


Figure 2.6: Histogram of Spending Across Various Categories

These histograms of spending across product categories reveal highly right-skewed distributions, where the majority of customers spend very little, while a smaller group of heavy spenders creates long tails. For example, wines and meat show the widest ranges, with spending extending beyond 1,400 and 1,700 units, respectively. This matches the earlier boxplots and the correlation matrix, where both categories were strongly tied to income (0.58 each) and stood out as premium items purchased by higher-income households.

By contrast, categories like fruits, fish, sweets, and gold have much lower spending overall, with most customers clustered under 50–100 units. Despite their smaller scales, these categories still exhibit long tails, suggesting the presence of niche but valuable customer groups who spend disproportionately more. The sharp peaks near zero across all six plots confirm that most customers are casual or low-frequency buyers, while the small group of high outliers accounts for a significant share of revenue.

These histograms of purchasing behavior show clear differences across the various channels customers use. Deal purchases are heavily skewed toward the low end, with most customers making only 1–3 discounted purchases, reinforcing the heatmap result that deals were only weakly correlated with income and product spending. Web purchases and catalog purchases also show right-skewed distributions, though with slightly higher spreads. While many customers make only a few purchases through these channels, a small group engages in repeated buying, especially via catalogs. This is consistent with the strong positive correlation between catalog purchases and income (0.59) as seen in the heatmap, suggesting that wealthier customers prefer catalogs for larger transactions.

Store purchases display the highest overall counts, with many customers clustering around 3–8 purchases. This aligns with the notion that in-store remains the most common method of buying, even if catalogs and web channels capture more high income spending. Website visits per month, however, show a different pattern: the distribution centers around 5–8 visits, indicating frequent browsing. But since income was negatively correlated with web visits (-0.55), this suggests that lower-income customers are driving much of this traffic without

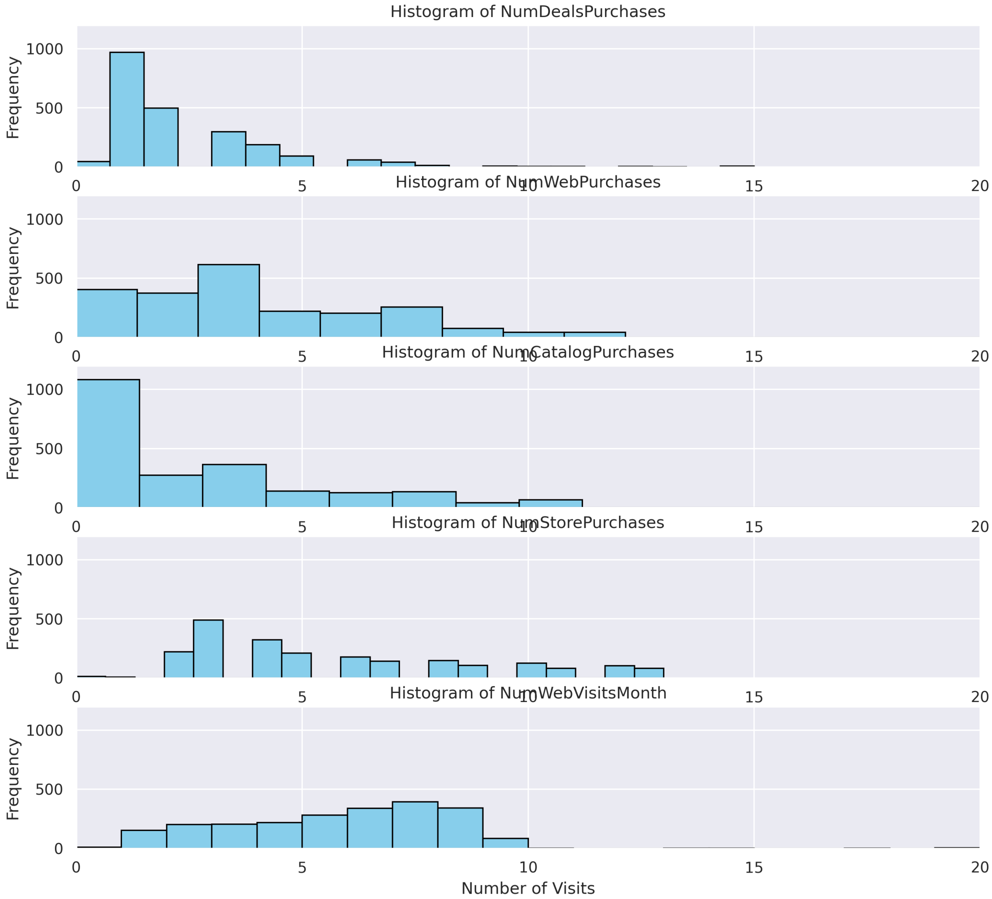


Figure 2.7: Histogram of customers different methods of purchases

necessarily converting into higher spending.

Overall, these patterns confirm the segmentation observed earlier: in store is the most widely used channel, catalogs are favored by higher income heavy spenders, and web visits are dominated by lowers pending customers who browse more frequently but purchase less.

Having examined the numerical variables and their relationships through descriptive statistics and visualizations, we now turn to the categorical features of the dataset to explore how demographic attributes and campaign responses shape customer behavior

2.2 Numerical Analysis and Graphical Analysis of Categorical Variables

After exploring the numerical variables and their relationships through descriptive statistics and visualizations, we now shift our focus to the categorical variables, examining how demographic characteristics, enrollment timing, and customer interactions influence spending behavior.

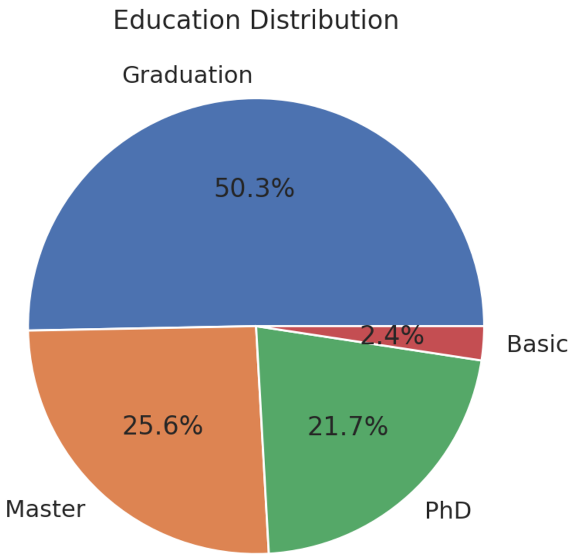


Figure 2.8: Proportion of Customers Education levels

Starting with education levels, the distribution highlights the academic background of the customer base and provides insight into how education may relate to income and spending habits. The education distribution shows that half of customers (50.3%) have completed undergraduate degrees (“Graduation”), while 25.6% have a Master’s degree and 21.7% hold a PhD. Only 2.4% report basic education. This skew toward higher education levels suggests that the customer base is generally well-educated, which may partially explain the higher spending observed in premium categories like wines and meat. In the correlation heatmap, education was not directly included, but since education typically correlates with income, this aligns with the strong positive correlations between income and wine (0.58) and meat

(0.58) spending.

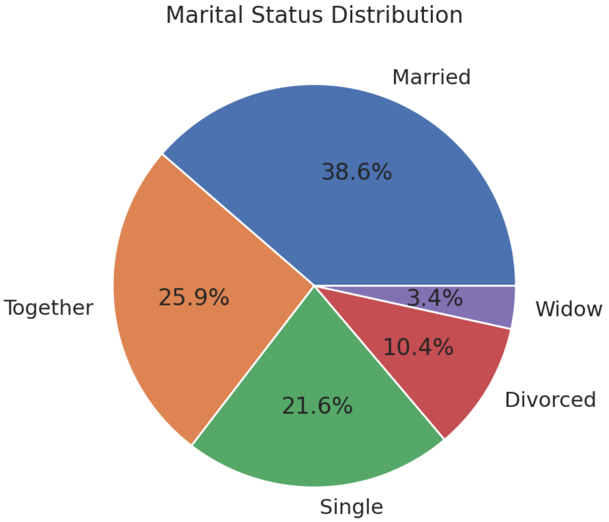


Figure 2.9: Proportion of Customers Marital Status

Building on demographics, marital status offers another important lens into household structure and its potential influence on purchasing behavior.The majority of customers are married (38.6%), followed by those living together (25.9%) and single (21.6%). Divorced (10.4%) and widowed (3.4%) groups make up smaller portions. These household structures can influence consumer behavior: for example, married or cohabitating customers may have higher family-related expenses, while single households may exhibit different purchasing priorities. This links back to the correlation matrix where Kidhome (-0.43 with income) and Teenhome (weak positive with deal purchases, 0.39) suggested family structure impacts not only income but also shopping style, such as deal-seeking or bulk purchases.

Shifting from static demographics to enrollment timing, the season distribution reveals how customer sign-ups were spread across different periods.The season distribution shows a relatively even spread of customer enrollment across 2013, with each season representing around 13% of the sample. Smaller proportions are seen in earlier 2012 (Winter and Summer) and later 2014 (Summer and Winter). This balance suggests the dataset avoids being dominated by one enrollment period, reducing temporal bias. Although seasonality itself does not appear in the correlation matrix, it could interact indirectly with recency — which

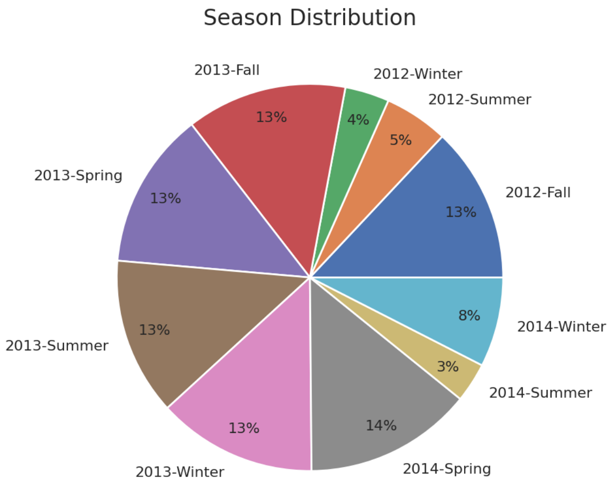


Figure 2.10: Proportion of customers in each season

the heatmap showed to have no strong correlation with other variables — suggesting that recency and seasonality are independent drivers of customer engagement.

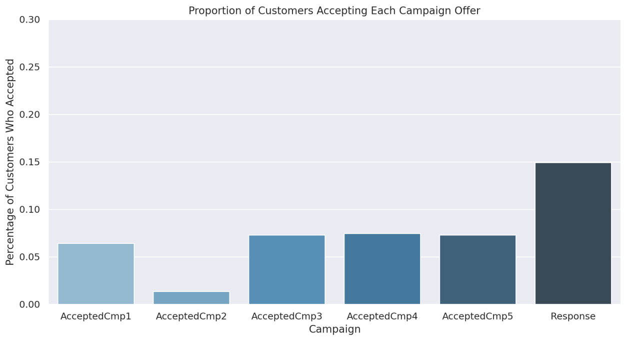


Figure 2.11: Proportion of customers accepting campaign offers

Beyond demographics and enrollment, customer interactions with marketing campaigns provide another key categorical variable for understanding engagement and responsiveness.Across the five campaigns, acceptance rates were relatively low, ranging from around 1% in Campaign 2 to about 7% in Campaigns 1, 3, 4, and 5. The final campaign response shows the highest acceptance, around 15%. This indicates that the latest marketing strategy was significantly more effective, possibly due to improved targeting. While campaign responses were not included in the correlation heatmap, they are crucial for segmentation: customers with higher incomes and stronger spending patterns (positively correlated with premium categories in the heatmap) may have been more likely to respond favorably, highlighting the role of targeted marketing.

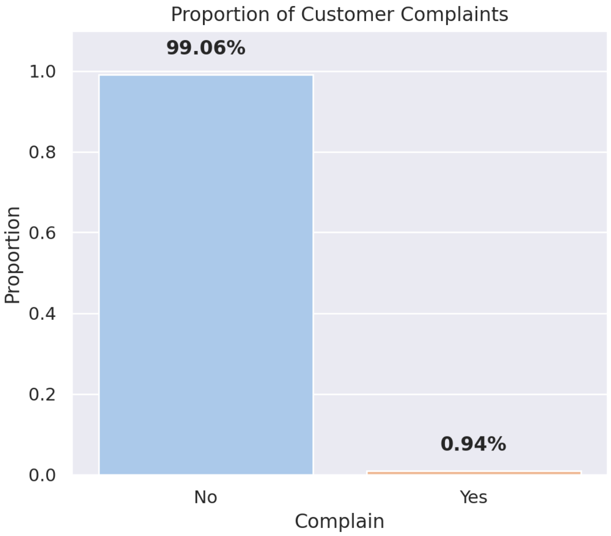


Figure 2.12: Proportion of Customers with Complaints

Finally, customer satisfaction is reflected in the complaints variable, which helps assess how negative experiences factor into overall purchasing behavior. Only 0.94% of customers lodged a complaint in the past two years, while over 99% reported no complaints. This suggests overall customer satisfaction is very high. Given the heatmap results, where recency showed little correlation with spending or demographics, complaints may not have been a strong driver of customer disengagement. Instead, factors like income and product preferences appear to shape spending much more than negative experiences. This finding reassures businesses that dissatisfaction is rare but emphasizes that maintaining satisfaction remains essential for retaining high-value customers, particularly those driving spending in premium categories.

3 Regression Analysis

In this section, we use linear and multiple linear regression to create models relating different covariates to our response variable, customer income. We begin by examining potential simple linear regression models. We then continue by exploring multiple linear models. In doing so, we create a "base" model containing all covariates we suspect may create a reliable model. Then, we use the data provided by this model to construct models using subsets of covariates. Next, we check for interactions between specific variables to confirm that there are no underlying interactions affecting the performance of our models. Through these processes, we find the model that uses the subset of covariates that will best predict customer income.

3.1 Linear Models

Because we are looking to reliably predict income, we need a model that successfully captures the relationship between the different factors. To do this, we must confirm that our factors conform to the model assumptions for linear regression: linearity, independence, normality, and equal variances. To further the accuracy of our data, we choose a subset of factors that we have found to have correlation with our response variable. These include the different categories of spending, number of website visits, and number of catalog purchases.

3.1.1 Simple Linear Regression

We first want to explore to what extent single variables can predict income. To start, we analyze the coefficient tables of a multitude of simple linear models. We then revisit the models' residual plots to identify any concerns regarding LINE assumptions. For our simple linear models, we analyse four variables that have high correlation with Income via figure 2.1: spending on wine and meat, number of website visits in a month, and number of catalog purchases. The linear models between Income and these factors are summarized in Table 3.1.

The R^2 values of each of these models are 0.4738, 0.5017, 0.5165, and 0.4275, respectively. We can see that all of these variables are statistically significant when modeled on their own, all having $\Pr(>|t|)$ less than 2e-16. However, the R^2 values help differentiate this picture,

Income MntWines	Estimate	Std. Error	t value	Pr(> $\lvert t\rvert$)
(Intercept)	38564.64	447.62	86.22	0
MntWines	43.88	0.9837	44.60	0
Income MntMeatProducts				
(Intercept)	40867.333	403.758	100.47	0
MntMeatProducts	6.652	1.455	47.17	0
Income NumCatalogPurchases				
(Intercept)	37698.3	33.4	86.99	0
NumCatalogPurchases	5371.1	110.5	48.59	0
Income NumWebVisitsMonth				
(Intercept)	82855.0	834.8	99.26	0
NumWebVisitsMonth	-5802.6	142.8	-40.63	0

Table 3.1: Simple Linear Regression: Spending Categories

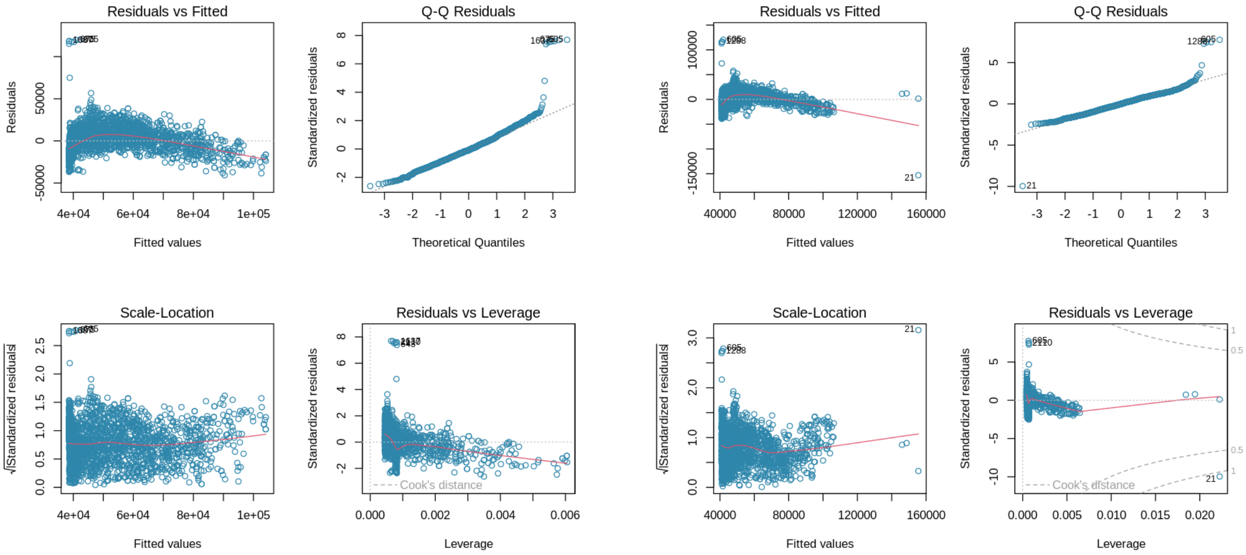
with meat spending and number of catlog purchases both higher than 50%. An interesting observation is that the coefficient of NumWebVisits is negative, signifying that customers who repeatedly view the store's website make less income than those who do not. Additionally, the standard error of MntWines is notably low, suggesting the coefficient is close to the true coeffient of the population. Now we move on to the number of kids and teens in the home. We choose to analyse these as we want to examine the potential economic factor that children play.

Income Kidhome	Estimate	Std. Error	t value	Pr(> $\lvert t\rvert$)
(Intercept)	61085.88	507.8070	120.29350	0
Kidhome	-20646.19	730.6235	-28.25832	0
Income Teenhome				
(Intercept)	51282.564	624.1864	82.159057	0.0000000
Teenhome	1359.253	840.3603	1.617464	0.1059206

Table 3.2: Simple Linear Regression: Household Composition

The R^2 values of these models are 0.265 and 0.001 for children and teens, respectively. This shows lower correlation than the previous categories of variables. This is reinforced by the low statistical significance shown by the number of teens in the home.

We have now seen the statistical significance and residual values of each of these variables. However, we need to ensure that these values conform to the assumptions for linear regression modeling. Figure 3.1 shows the residual plots of wine and meat, which were the two models from spending categories with the most promise.



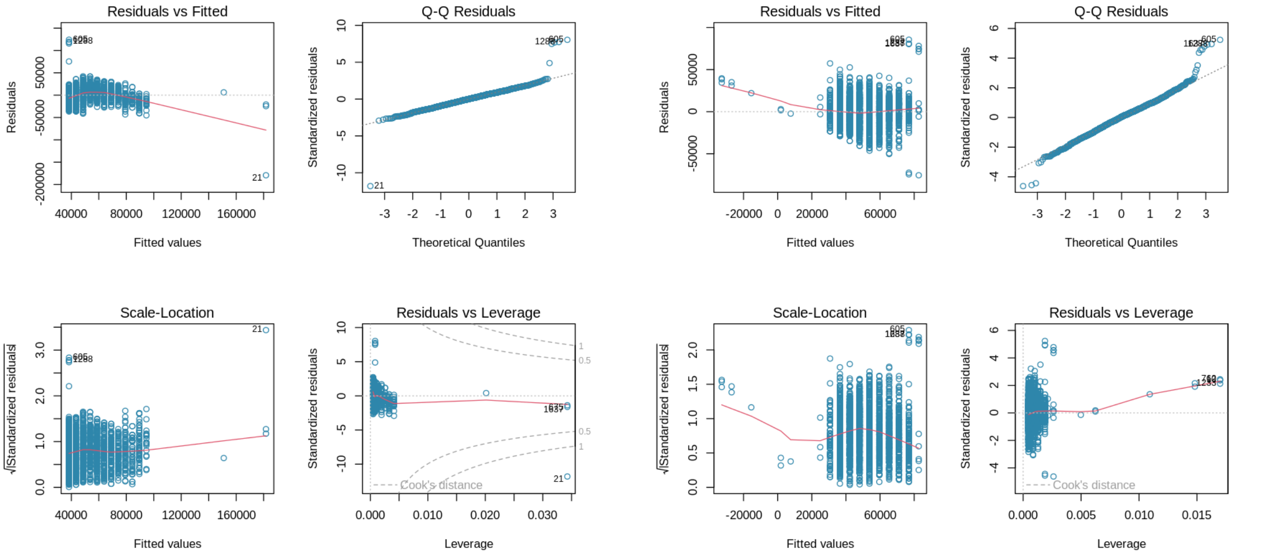
(a) Residual Plots of Income MntWine

(b) Residual Plots of Income MntMeatProducts

Figure 3.1: Residual Plots of Spending Categories

In figure 3.1, the wine model's residuals demonstrate fairly consistent spread and thus homoscedasticity, an approximately normal distribution with the Q-Q plot showing overall alignment with the theoretical line, and a fairly uniform spread. There are a few high-leverage points, but they are less severe than what can be observed in the meat model's leverage plot. The meat model also shows a slight funneling pattern, indicating slight heteroscedasticity. Its Q-Q plot is similar to the wine model's, with overall alignment with the theoretical line but some departure at the extreme ends. The meat model's leverage graph also shows points of concern close to the line of Cook's distance, indicating points that may heavily influence the model.

Figure 3.2 shows the residual plots resulting from the linear models of income predicted by number of catalog purchases and income predicted by number of website visits. Both of these sets of plots show divergence from the LINE assumptions for linear regression models. This is particularly due to the discrete nature of these variables, with both variables' largest



(a) Residual Plots of Income NumCatalogPurchases

(b) Residual Plots of Income NumWebVisitsMonth

Figure 3.2: Residual Plots of Purchasing Channels

observations being less than 30, potentially limiting their predictive power.

3.2 Multiple Linear Models

Now that we have explored simple linear models, we move on to our exploration of multiple linear regression models. We choose to exclude the campaign response variables due to their responsive nature. Whether or not a customer responds to a marketing campaign is more likely predicted by income than predictive of income. Additional variables, such as cost of customer contact, were also excluded for similar reasons. Redundant binary data could also add unnecessary complexity. We start by constructing a base model using the following 16 variables, chosen by their plausible relation to income:

We determine that Teenhome, MntWines, MntMeatProducts, NumWebPurchases, NumCatalogPurchases, and NumWebVisitsMonth are statistically significant in this model. It is notable that the four variables with the highest correlation to Income are part of this subset. We also find an R^2 value of 0.7794, an adjusted R^2 value of 0.7773, and a residual standard

	Estimate	Std. Error	t value	Pr($> \|t\|$)
(Intercept)	50256.450	7376.772	6.813	1.23e-11
factor(Education)Basic	-10448.429	1581.847	6.605	4.97e-11
factor(Education)Graduation	758.418	786.927	0.964	0.335267
factor(Education)Master	1255.406	913.429	1.374	0.169462
factor(Education)PhD	2146.400	893.553	2.402	0.016384
factor(Marital_Status)Divorced	-2888.356	7300.836	-0.396	0.692424
factor(Marital_Status)Married	-3433.096	7274.632	-0.472	0.637026
factor(Marital_Status)Single	-3840.489	7280.916	-0.527	0.597918
Kidhome	1975.078	545.277	3.622	0.000299
Teenhome	5895.962	475.663	12.395	¡ 2e-16
Recency	-10.397	7.483	-1.389	0.164873
MntWines	14.818	1.047	14.154	¡ 2e-16
MntFruits	13.510	7.567	1.786	0.074312
MntMeatProducts	22.138	1.661	13.328	¡ 2e-16
MntFishProducts	4.165	5.750	0.724	0.468982
MntSweetProducts	27.121	5.138	-1.378	0.000204
MntGoldProds	-7.078	5.138	-1.378	0.168474
NumDealsPurchases	-515.367	145.162	-3.550	0.000393
NumWebPurchases	999.450	109.659	9.114	¡ 2e-16
NumCatalogPurchases	1018.332	130.341	7.813	8.61e-15
NumStorePurchases	378.049	101.500	3.725	0.000201
NumWebVisitsMonth	-2983.893	129.025	-23.127	¡ 2e-16

Table 3.3: Multiple Linear Regression: Base Model with 16 Variables

error of 10,150. This means that approximately 77% of the variation is explicable by the model, after accounting for the number of terms. It is known that the R^2 value will always increase with the introduction of additional features. So, we will prune this model to find a smaller subset of features that can approach similar residual standard error and R^2 values.

In the model's residual plots, we find minor points of concern. While the Residuals vs. Fitted plot shows linearity and the Q-Q plot shows the points following a normal pattern, the Scale-Location plot shows noise with one potential outlier. The Residuals vs. Leverage plot similarly shows potential high-leverage points.

With this baseline in mind, we choose new subsets of variables to explore. For our first subset, we examine customers' spending across different categories. Within the five spending categories, we see that wine and meat are clearly significant variables, with sweets showing

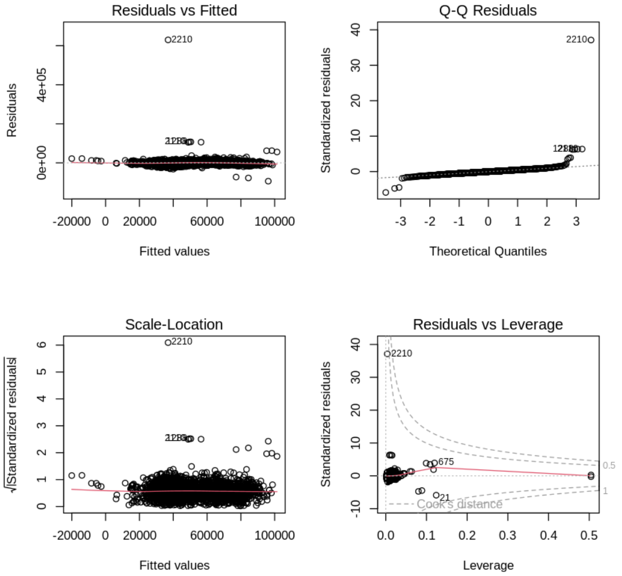


Figure 3.3: Residual Plots of Multiple Linear Regression Base Model

significance to a lesser degree. We form a multiple linear regression model with the five spending variables and find that spending on gold products is insignificant, so we remove it from the model to obtain Model 1, summarized in Table 3.4.

	Estimate	Std. Error	t value	Pr(> $\|t\|$)
(Intercept)	35482.67500	394.772804	89.881255	¡ 2e-16
MntWines	26.39530	1.017878	25.931696	¡ 2e-16
MntFruits	54.17624	9.234538	5.866698	¡ 5.01e-08
MntMeatProducts	33.22402	1.768062	18.791203	¡ 2e-16
MntFishProducts	38.50507	6.852002	5.619535	3.27e-07

Table 3.4: Multiple Linear Regression: Model 1 Summary

Looking at the model summary in table 3.4, we can see that wine and meat product categories are the most significant. However, sweets, fruits, and fish also show some level of significance. This model has an R^2 value of 0.6338, an adjusted R^2 value of 0.6332, and an RSE of 13,030. This marks a decrease of explicability of variation and an increase in error compared to our base model. We note that spending on wine and meat maintain significantly small p-values, as do fish and fruits to a lesser degree. In Figure 3.4, we see trends of homoscedasticity and normality. There are some potentially high-leverage points,

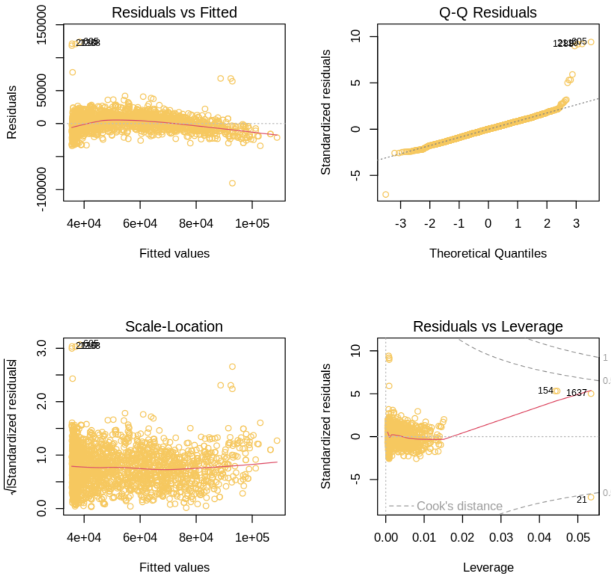


Figure 3.4: Multiple Linear Regression Model 1 Residuals

but nothing of extreme concern.

We test a second multiple linear regression model (Model 2). In this model, we use the four features the most correlated to Income, which we noted as all statistically significant in our base model. Model 2 is summarized in Table 3.5. It has an R^2 and adjusted R^2 value of 0.7286, and an RSE of 11,220. These values show further improvement in predictive capability compared to Model 1, but remain marginally worse than our base model. The residuals show similar adherence to LINE assumptions compared to previous models, with a mostly normal Q-Q plot, some noise present in the Scale-Location plot, and a couple of potential high-leverage points in the leverage plot.

	Estimate	Std. Error	t value	Pr(> $\|t\|$)
(Intercept)	55060.6253	852.5051	64.59	¡ 2e-16
MntWines	2.7904	0.9531	23.91	¡ 2e-16
MntMeatProducts	19.0570	1.6672	11.43	¡ 2e-16
NumCatalogPurchases	1205.5327	136.3679	8.84	¡ 2e-16
NumWebVisitsMonth	-3084.7342	120.5364	-25.59	¡ 2e-16

Table 3.5: Multiple Linear Regression: Model 2 Summary

Our third multiple linear regression model (Model 3) adds the Teenhome variable. This

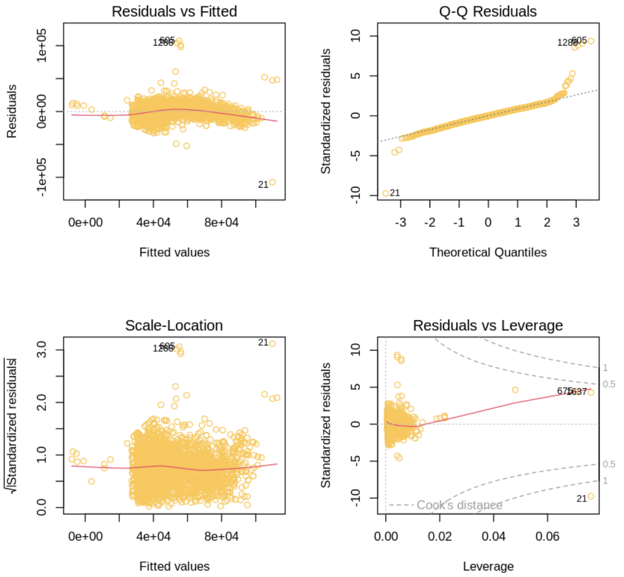


Figure 3.5: Multiple Linear Regression Model 2 Residuals

variable was significant in our base model, so we attempt to add it to Model 2 to seek further performance gains. In Table 3.6, we can see all of the covariates are statistically significant. This model gives an R^2 value of 0.7527 and an adjusted R^2 value of 0.7521, which shows another improvement over both Model 1 and Model 2. When we compare these values to those of our base model, we find extremely marginal differences between RSE and R^2. Our base model has an adjusted R^2 value and RSE of 0.7773 and 10,150 while Model 3's values are 0.7521 and 10,710, respectively. Model 3 obtains these values from only five features, while our base model uses 16. Once again, the residual plots demonstrate similar levels of adherence to LINE assumptions, as seen in Figure 3.6.

	Estimate	Std. Error	t value	$\Pr(>\|t\|)$
(Intercept)	52563.96981	865.9792647	60.698878	0
MntWines	22.58923	0.9281671	24.337456	0
MntMeatProducts	24.32901	1.6963610	14.341883	0
NumCatalogPurchases	850.47086	131.3002531	6.477298	0
NumWebVisitsMonth	-3200.93819	117.6364491	-27.210428	0
Teenhome	6316.75263	454.0247780	13.912793	0

Table 3.6: Multiple Linear Regression: Model 3 Summary

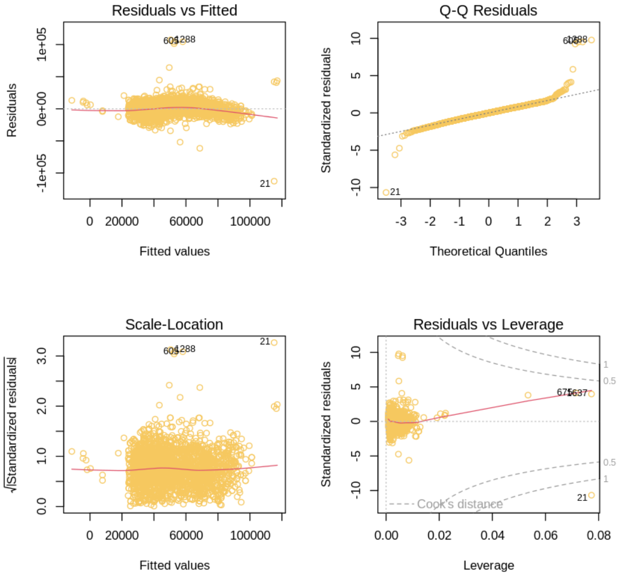


Figure 3.6: Multiple Linear Regression Model 3 Residuals

3.2.1 Model Comparison Using ANOVA

We use ANOVA tables to compare these three models. Tables 3.7, 3.8, and 3.9 show the results. Table 3.7 draws comparisons between Models 1 and 2. These two models share the same degrees of freedom, so the ANOVA does not perform well between them. It does show a decrease in RSS, but is unable to display the F-statistic and does not carry much weight.

The ANOVA does, however, show significant improvement between Models 2 and 3, with the RSS decreasing and an extremely small p-value (3.05e-42). This means the difference between the two models is statistically significant. This also holds true for the comparison between Models 1 and 3, which shows an even more dramatic difference (p-value of 6.73e-176). Based on these tables, we can establish that Model 3 is the best performing model, while Model 1 is the least effective.

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
2208	383417295746	NA	NA	NA	NA
2208	290275640748	0	93141654998	NA	NA

Table 3.7: ANOVA: Model 1 vs. Model 2

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
2208	290275640748	NA	NA	NA	NA
2207	266869726564	1	23405914184	193.56583	3.050137e-42

Table 3.8: ANOVA: Model 2 vs. Model 3

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
2207	266869726564	NA	NA	NA	NA
2208	383417295746	-1	-116547569182	963.84296	6.725019e-176

Table 3.9: ANOVA: Model 3 vs. Model 1

3.3 Interaction Model

Following the examination of the multiple linear models, it was clear that Model 3 provided the best fit to predict income. This model included the covariates: MntWines, MntMeatProducts, NumCatalogPurchases, NumWebVisitsMonth, and Teenhome. In table 3.10, we test the interaction between two variables that make logical sense in the context of european consumers: spending on wine and meat. We hypothesize that there exists a dependence between how customers spend on wine and meat in proportion to their income. From table 3.10, we see that the interaction term has a coefficient of -0.0741, while the separate terms have coefficients of 79.07 and 41.81 for meat and wine respectively. This is intriguing as a negative coefficient suggests that spending more on each of these items

	Estimate	**Std. Error**	**t value**	**Pr($>|t|$)**
(Intercept)	3.304+e04	3.943e+02	83.81	¡ 2e-16
MntMeatProducts	7.907e+01	2.224e+00	35.55	¡ 2e-16
MntWines	4.181e+01	1.211e+00	34.52	¡ 2e-16
MntMeatProducts:MntWines	-7.410e-02	3.712e-03	-19.96	¡ 2e-16

Table 3.10: Interaction Model: MntMeatProducts and MntWines

Here we can see that all predictors in this interaction model are statistically significant (p-values approximately zero) and contribute to predicting the subjects' income. The interaction term being well above zero indicates that the negative effects of web visits are less

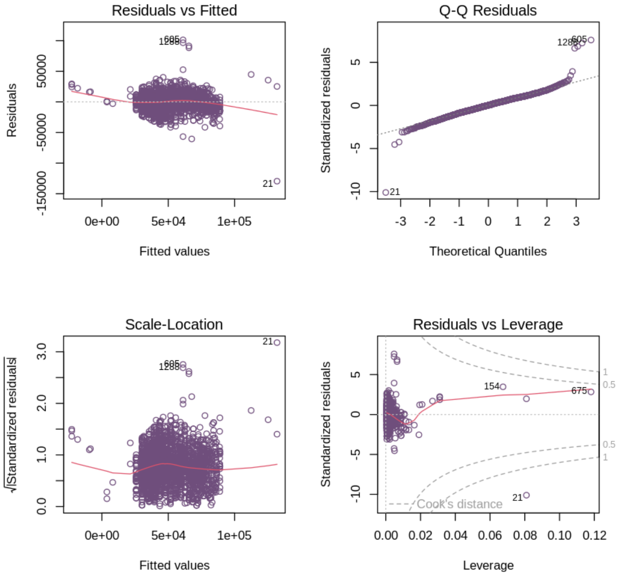


Figure 3.7: Interaction Model Residuals

severe when catalog purchases are high. However, there are a few concerns with the plots of this interaction model. There are slight deviations from the diagonal on the Q-Q plot and some small curvature in the Residuals vs. Fitted plot, which would indicate a violation of homoscedasticity. The model also has an adjusted R^2 value of 0.4834, which indicates that despite the significant p-values, this interaction model does not perform as well as our original Model 3.

3.4 Log-Transformed Model

To accommodate for the slight skew in our best model (Model 3), we applied a log transformation, which yielded the following results:

From the table, we see that most p-values are very close to zero, indicating that most covariates are significant in predicting the subjects' income. Note that NumCatalogPurchases_scaled has a p-value of 0.1348, suggesting it may not be significant after transformation. Additionally, each residual plot appears to follow the LINE assumptions with little to no violations. This is a better indication of model quality than the interaction model.

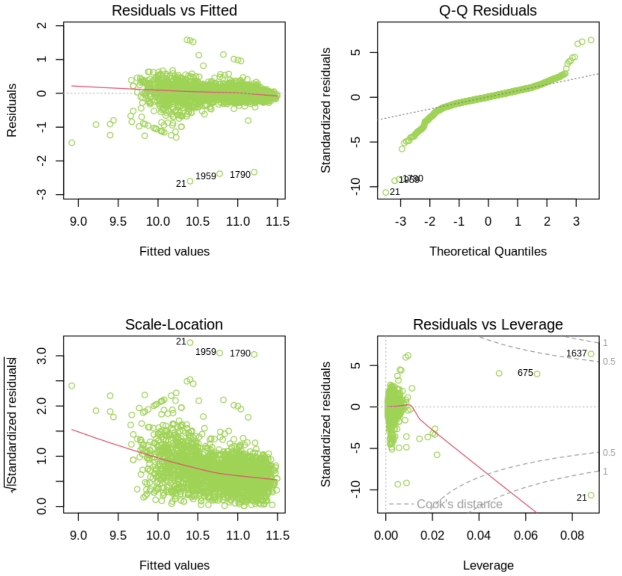


Figure 3.8: Log-Transformed Model Residuals

Table 3.11: Log-Transformed Model: Regression Coefficients Summary

	Estimate	**Std. Error**	**t value**	**Pr($>|t|$)**
(Intercept)	9.8271	0.0236	416.9071	0.0000
MntWines_log	0.1586	0.0061	26.0766	0.0000
MntMeatProducts_log	0.0346	0.0079	4.3905	0.0000
NumCatalogPurchases_scaled	-0.0127	0.0085	-1.4960	0.1348
NumWebVisitsMonth_scaled	-0.1783	0.0065	-27.3152	0.0000
Teenhome	0.0803	0.0114	7.0407	0.0000

Finally, this model has an R^2 value of 0.7414, which represents a good fit and is comparable to Model 3's performance of 0.739.

3.5 Summary

In this analysis, we employed both simple and multiple linear regression to model customer income using various behavioral and demographic covariates. Initial simple linear models showed that spending on wines and meat products had the strongest individual relationships with income, supported by higher R^2 values and relatively well-behaved residuals. In contrast, website visits and catalog purchases exhibited weaker fits, with website visits showing a negative association with income. Building on these insights, we developed a series of multiple regression models. The base model revealed that variables such as MntWines, MntMeatProducts, NumCatalogPurchases,NumWebVisitsMonth, and Teenhome were statistically significant, achieving an adjusted R^2 of 0.544. Refining this model through variable selection led to substantial improvements, with the best-performing model (Model Three) including MntWines, MntMeatProducts, NumCatalogPurchases, NumWebVisitsMonth, and Teenhome, and reaching an adjusted R^2 of 0.739. Residual diagnostics confirmed reasonable adherence to linear regression assumptions, with only minor issues related to heteroscedasticity and leverage. Finally, an interaction model between NumCatalogPurchases and NumWebVisitsMonth revealed a significant moderating effect, indicating that frequent catalog purchasers are less negatively affected by increased web visits. However, despite its significance, this interaction model performed worse overall R^2 of 0.4834 compared to the best multiple regression model, confirming that the latter provides the most reliable and interpretable prediction of customer income.

4 Classification Analysis

In this section, we shift from predicting continuous income values to classifying customers into High Income and Low Income groups. This classification task provides insight into how effectively the key behavioral and demographic variables identified earlier particularly spending on wine and meat products, web engagement, and household structure can distinguish between income levels. To conduct this analysis, we implement a broad range of supervised learning models, including linear classifiers, distance-based methods, probabilistic approaches, and ensemble-based techniques. Each classifier is evaluated under three training strategies: the Full Dataset, a 50% Validation Split, and 5-Fold Cross Validation. These approaches allow us to assess not only predictive accuracy but also the stability and generalizability of each model. Performance is measured using accuracy, sensitivity, specificity, and runtime, with special emphasis on correctly identifying high-income customers, the class of primary interest for targeted marketing and business decision-making. The following subsections detail the setup, results, and comparative performance of each classifier, ultimately identifying the most robust and effective models for income-level prediction.

4.1 Model Setup

In this classification analysis, our objective is to categorize customers into two groups: Low Income(0) and High Income(1). The predictors selected for this task are the same variables identified as most influential in the regression analysis, including `MntWines`, `MntMeatProducts`, `NumWebVisitsMonth`, and `Teenhome`. These features capture key spending behaviors and household characteristics that are expected to differ across income levels.

4.2 Logistic Regression

Logistic regression is a classification technique that uses a special function, $\sigma(z)$, to produce a probability between zero and one. This function shown in figure 4.1, called a sigmoid function, uses the weighted sum of input features to produce this output. The weights of each input feature w_i and the bias b are determined by the minimization of the cross entropy (log loss) function. The cross entropy equation enforces choosing weights that maximize separation between classes within the sigmoid function. The sigmoid function returns a probability between zero and one. This probability is compared with a set threshold value, typically 0.5, to determine the class to which a sample belongs. Probabilities higher than 50% indicate one class, while probabilities less than 50% indicate the other.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$z = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b$$

Figure 4.1: The sigmoid function used in logistic regression.

We applied 5-fold cross validation to evaluate our logistic regression model. We split our dataset into five randomized discrete subsets. On each fold, we use four of the folds for model training, while the fifth is used for testing the model. A different subset is used for training on each fold. The ROC curves for each of the five models is modeled in figure 4.2. The models' precisions, specificities, and overall accuracies are described in table 4.1.

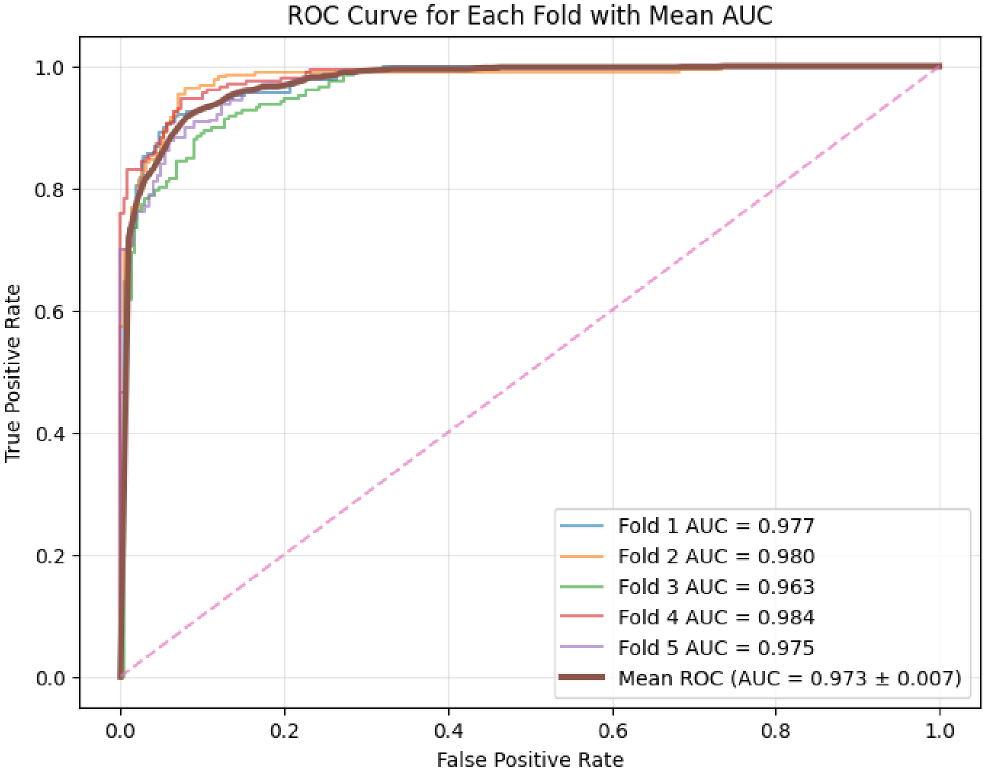


Figure 4.2: The ROC curves of five logistic regression models evaluated via 5-fold cross validation

Logistic Regression											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.913	0.907	0.914	0.895	0.864	0.899	0.931	0.952	0.929	0.040	0.037	27.914

Table 4.1: Accuracy, sensitivity, specificity, and run time for Logistic Regression using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

With the data gathered from table 4.1, we see that all accuracies gathered from the logisitc regression method are above 0.900 with minor differences between the three values. We also take note that the specificity values in all of the three models is higher than that of the three values of sensitivity. This indicates that logisitc regression is more tenative to label customers as High Income. Regarding run time, naturally the cross-validation model had a run time well above the full and split data models' run times, with the cross-validation model having the longest run time in the whole study. Looking deeper, we used a Reciever Operation Characteristic (ROC) curves to further observe if logistic regression is a fair predictor of Income. To acheive this we apply the ROC to each of the five folds in our cross-validation of logistic regression. From figure 4.2, we see that each of the five curves bunch together near the top left corner of the graph. It is also seen that the average Area Under the Curve (AUC) is 0.973 with a standard deviation of 0.007. These two observations, along with the high accuracy values, logisitc regression serves as a strong predictior of Income levels. However, with the high run time of the cross-validation model, logisitc regression could prove to not be the best classifier.

4.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a generative classification method that models the joint distribution of the predictors within each class and then applies Bayes' rule to obtain class probabilities. In contrast to logistic regression, which directly models the conditional probability $P(Y = 1 \mid X)$, LDA assumes that the predictors X follow a multivariate normal distribution within each income group and that both groups share a common covariance matrix.

Let μ_k denote the mean vector of the predictors in class $k \in \{0, 1\}$, let Σ be the common covariance matrix shared across classes, and let π_k be the prior probability of belonging to class k. Under the LDA model, the classifier assigns an observation with feature vector x to the class with the largest discriminant score:

$$\delta_k(x) = x^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log(\pi_k).$$

The first term measures how aligned x is with the class mean after adjusting for the covariance structure, while the remaining terms penalize classes with less favorable location or smaller prior probability. Because high-income customers tend to have larger values of MntWines, MntMeatProducts, and NumCatalogPurchases and fewer NumWebVisitsMonth, the estimated μ_1 vector shifts the discriminant function so that profiles with strong premium spending and lower web engagement are more likely to be classified as High Income.

We evaluate LDA on the following resulting accuracies, sensitivities, specificities, and runtimes that are summarized in Table 4.2. Across all three setups, LDA achieves overall accuracies close to 0.900, with only minor differences between the full-data, 50% split, and cross-validation approaches. As in the regression analysis, the classifier tends to favor correct identification of Low Income customers: specificity remains higher than sensitivity in each setting, indicating that LDA is slightly more conservative when labeling customers as High Income.

LDA											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.897	0.900	0.896	0.849	0.850	0.848	0.946	0.949	0.945	0.048	0.031	2.413

Table 4.2: Accuracy, sensitivity, specificity, and run time for LDA using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

To further assess threshold–independent performance, we compute the ROC curves for each of the five cross-validation folds. These curves are shown in Figure 4.3, along with the mean ROC curve. The individual fold curves cluster tightly near the top-left corner of the plot, and the mean AUC is approximately 0.970 with a standard deviation of about 0.009. This high and stable AUC indicates that, even when the decision threshold is varied, LDA consistently separates High Income and Low Income customers very well. Combined with its relatively low runtime, these results suggest that LDA provides a strong linear benchmark for income-level classification, performing comparably to logistic regression while relying on a different modeling framework.

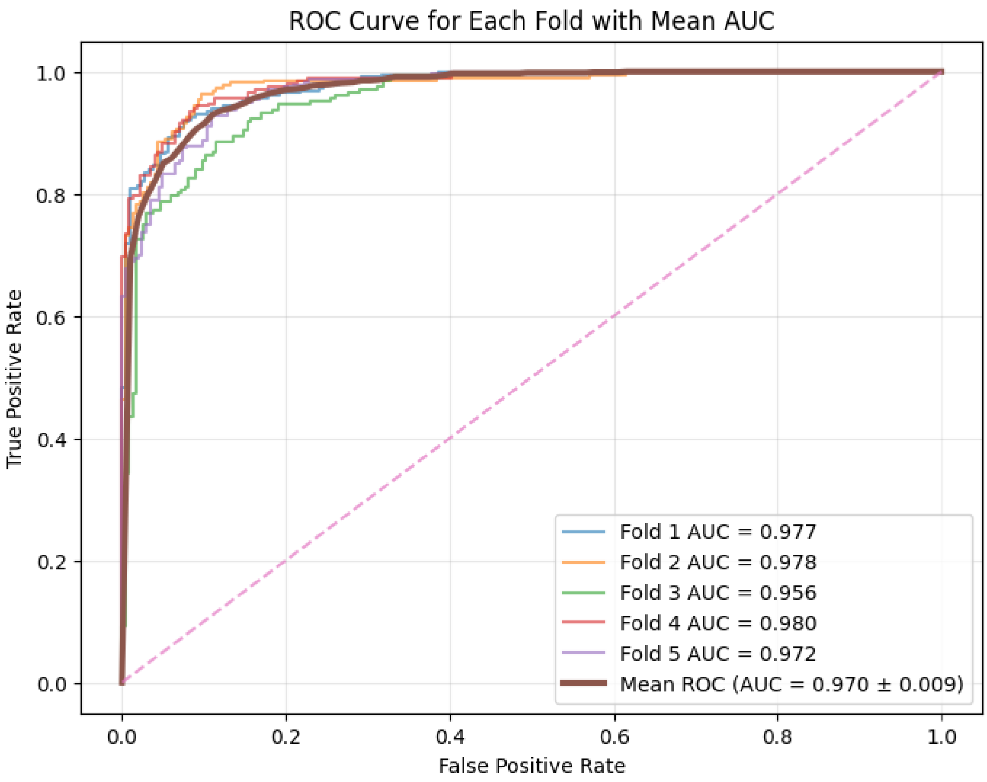


Figure 4.3: ROC curves for the five LDA models evaluated via 5-fold cross validation, with mean ROC and mean AUC of 0.970 ± 0.009.

4.4 Qualitative Data Analysis

Quadratic discriminant analysis (QDA) generalizes LDA by allowing each class to have its own covariance matrix. This leads to a quadratic decision boundary that can adapt to more complex relationships between the predictors and the response. QDA is particularly useful when the variability or correlation structure differs substantially across classes. In the consumer-profile dataset, high-income customers not only spend more in several categories but also exhibit greater variability in some of these expenditures. Allowing class-specific covariance matrices may better capture the shape of the high-income cluster in the predictor space and improve classification for observations with more extreme spending patterns.

Let μ_k denote the mean vector of class k, Σ_k the covariance matrix for class k, and π_k

the prior probability of class k. The QDA discriminant function is

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k) + \log(\pi_k),$$

and a new observation is classified into the class with the largest value of $\delta_k(x)$. The quadratic term $(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)$ allows the decision boundary to bend around regions where one class is more dispersed than the other. Because the high-income group tends to display more variability in premium spending, we expect QDA to potentially increase sensitivity relative to LDA by better capturing these curved boundaries. However, QDA estimates more parameters and can be less stable when class sizes are modest or when outliers are present, so performance gains are not guaranteed for every dataset.

The results of the QDA, in Table 4.3 indicate that this model achieves consistently strong overall accuracy across the full dataset, 50%, and cross-validation methods with the values remaining close to 0.900 in each case. Unlike LDA, QDA allows each class to have it's covariance structure, meaning it directly accounts for how spending variables such as wine,meat,catalog purchaes, and web purchases vary together within each customer group. This added flexibility makes QDA more conservative in assigning customers to the high-income class, which is reflected in its consistently higher specificity than sensitivity. While this reduces its ability to capture every high-income customer, it increases confidence in the customers it does classify as higher income. QDA also maintains a relatively low runtime, making it an efficient nonlinear classification option for our dataset.

QDA											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.896	0.892	0.893	0.832	0.852	0.829	0.959	0.931	0.957	0.069	0.041	2.498

Table 4.3: Accuracy, sensitivity, specificity, and run time for QDA using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

Looking at the ROC curve for the QDA classifier, we once again notice the pattern of the curves clustering together towards the top left corner of the graph. In comparison to that of the logistic regression and LDA classifiers, the third fold specifically is not as close to the corner, but is still an adequate fit. All of the AUC values exceed 0.950, with a mean AUC of 0.970 and a standard deviation of 0.009. Taking all that into account along with the data gathered from the summary of the QDA, while a fine classifier for predicting Income, it still performs worse than that of the logisitc regression and LDA classifiers.

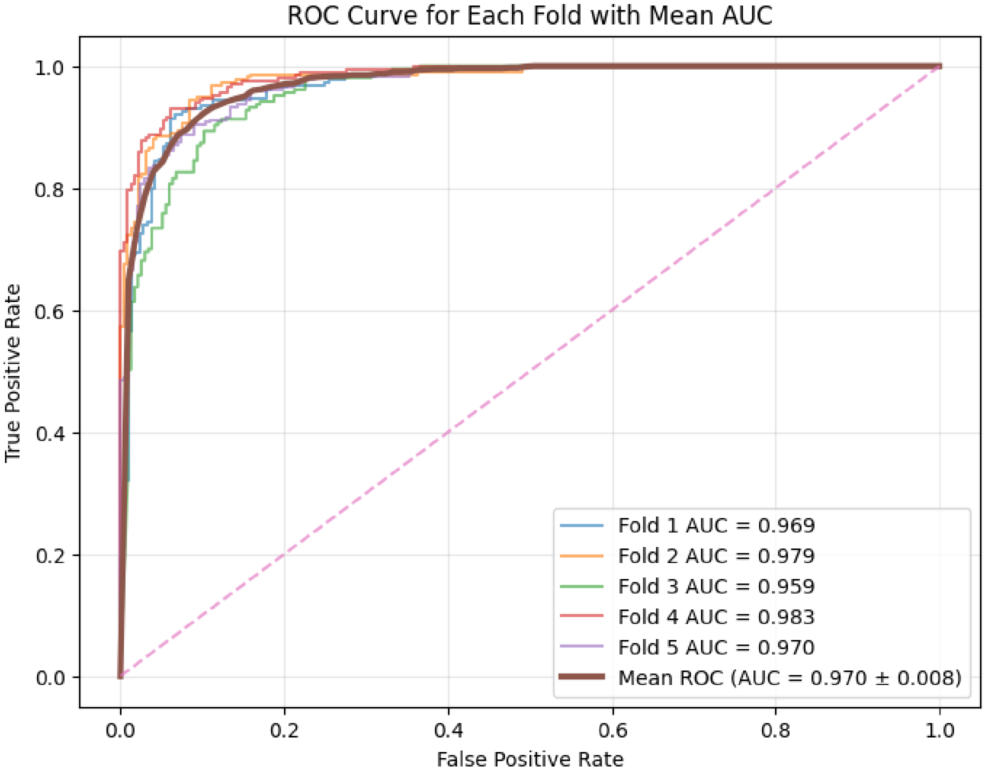


Figure 4.4: The ROC curves of five QDA models evaluated via 5-fold cross validation

4.5 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a learning algorithm for classification and regression that uses an unknown point's proximity to known data points to predict its value. For each new data point, the k closest labeled data points are found via a distance function, such as Euclidean distance:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

This formula measures the distance of two points on a plane. For classification, the simple majority of the k nearest neighboring points determine the predicted class. When using regression, the average value of these k nearest points is the predicted value.

The KNN algorithm is sensitive to the choice of k. When k is small, the model becomes sensitive to variation in data. This choice of k may lead to overfitting, when the model becomes too specific to the training data. If the chosen k is too large, the model may suffer from higher bias and underfitting, where the model can't accurately make predictions due to its simplicity. This model is also sensitive to different scales of the input features. Features with large scales may have a higher affect on the prediction than features of smaller scales, such as the difference in scale between income (0-100,000+) and age (0-100).

	KNN											
	Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Folds	Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
k = 1	0.999	0.863	0.879	0.999	0.857	0.891	0.999	0.868	0.867	0.134	0.068	1.263
k = 2	0.947	0.850	0.865	0.894	0.774	0.806	1	0.927	0.925	0.202	0.221	1.438
k = 3	0.931	0.865	0.882	0.938	0.868	893	0.923	0.862	0.872	0.119	0.128	1.004
k = 4	0.919	0.869	0.880	0.894	0.827	0.851	0.944	0.912	0.909	0.161	0.072	0.751
k = 5	0.914	0.880	0.879	0.902	0.899	0.890	0.907	0.861	0.867	0.306	0.143	0.825
k = 6	0.911	0.874	0.877	0.891	0.854	0.866	0.930	0.894	0.888	0.106	0.082	0.830
k = 7	0.903	0.882	0.885	0.918	0.903	0.897	0.888	0.862	0.872	0.324	0.075	0.668
k = 8	0.903	0.888	0.881	0.898	0.885	0.877	0.907	0.891	0.885	0.180	0.100	1.565
k = 9	0.902	0.893	0.884	0.917	0.919	0.899	0.887	0.867	0.868	0.107	0.123	0.682
k = 10	0.907	0.896	0.884	0.896	0.908	0.883	0.899	0.883	0.885	0.179	0.092	0.721

Table 4.4: Accuracy, sensitivity, specificity, and run time for KNN (k=1 to k=10) using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

To elaborate upon table Table 4.4, we see accuracies above 0.900 for all of the folds when all of the data is being used. When looking at the split and the cross-validation data we see the accuracies decrease slightly into the range of (0.850 - 0.890). It is also important to note that the specificity remains higher than the sensitivity, meaning that it is easier for the KNN classifier to correctly predict customers with low income. After examining the 10 differnt k-values, we found that the optimal k-value based on the four metrics depicted in the table, was k = 7. The ROC curve for **k=7**, shown in Figure 4.5, further reinforces this conclusion. The curve remains close to the top-left corner of the plot, reflfecting strong overall classification performance. The consistently high area under the curve (AUC) demonstrates that this model effectively distinguishes between income groups across a wide range of decision thresholds.

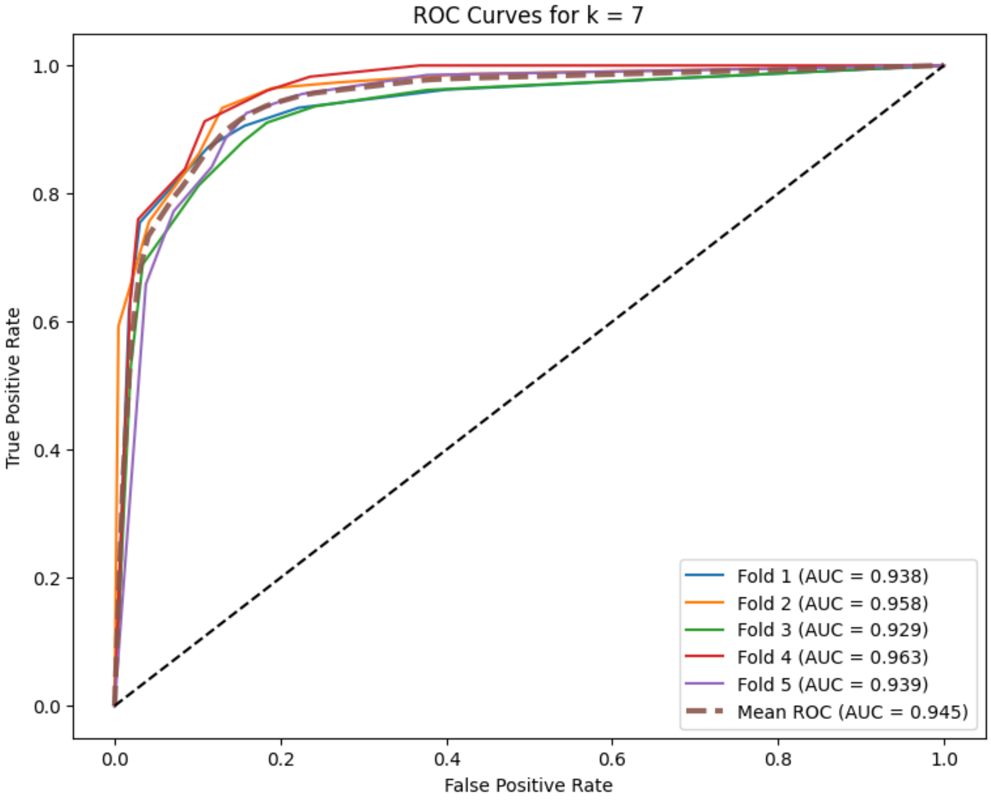


Figure 4.5: The ROC curves of five KNN at k = 7 models evaluated via 5-fold cross validation

4.6 Naive Bayes

The Naive Bayes classifier is a probabilistic classifier that uses basic Bayesian probability rules to calculate the probability of each feature given each class. The probabilities of each feature are then used to find the posterior probability for each class given the set of observed features. The final prediction is found using maximum likelihood of each class. That is, the class with the highest total probability is used as the prediction. In mathematical terms, Bayes classifier is represented by the equation:

$$C^{Bayes}(x) = \arg\max P(Y = r|X = x)$$
$$r \in \{1, 2, 3, ..., K\}$$

This classifier makes several assumptions, the first being that all features are independent of one another. From this *naive* assumption this classifier earns its name. The second and third assumptions are that continuous features are normally distributed and discrete features have multinomial distributions. Naive Bayes also assumes that all features are equally important and no data is missing.

The results for the Naive Bayes classifier, summarized in Table 4.5, reflects the strengths and limitations of a probability-based method that relies on simplifiying assumptions. Across all three training strategies, with accuracies of 0.894 using the full dataset, 0.892 under the 50% split, and 0.894 under 5-fold cross validation. Sensititivty remains lower than specificity in every case, with cross-validation values of 0.835 and 0.954, respectively. This indicates that the model is far more confident when identifying customers as low-income(0) than when detecting higher-income(1) ones. This outcome aligns with the structure of Naive Bayes, which assumes that each spending variable contributes independetly to the final decision. While this assumption limits the model's ability to capture interactions between features such as wine,meat and catalog purchases, it also helps prevent overfitting and promotes stable performance across data splits. The runtime remains low across all training strategies, with the cross-validation model completing in 3.760 seconds, reinforcing Naive Bayes as an efficient classification approach.

Naive Bayes											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.894	0.892	0.894	0.833	0.856	0.835	0.954	0.928	0.954	0.051	0.036	3.760

Table 4.5: Accuracy, sensitivity, specificity, and run time for Naive Bayes using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

The ROC curves for Naive Bayes, displayed in Figure 4.6, illustrate this balance between simplicity and performance. The curves remain well above the diagonal reference line, showing the classifier performs substantially better than random guessing across a wide range of thresholds.Even as the threshold moves away from the default 0.5 cutoff used to obtain the sensitivity of 0.835 and specificity 0.954 in cross-validation, the classifier preserves a strong trade-off between correctly detecting higher income customers and avoiding false alarms among lower income customers. The shape and positioning of these curves suggest a high and relatively stable AUC, reinforcing Naive Bayes as a fast, numerically reliable baseline for distinquishing between two customer income levels.

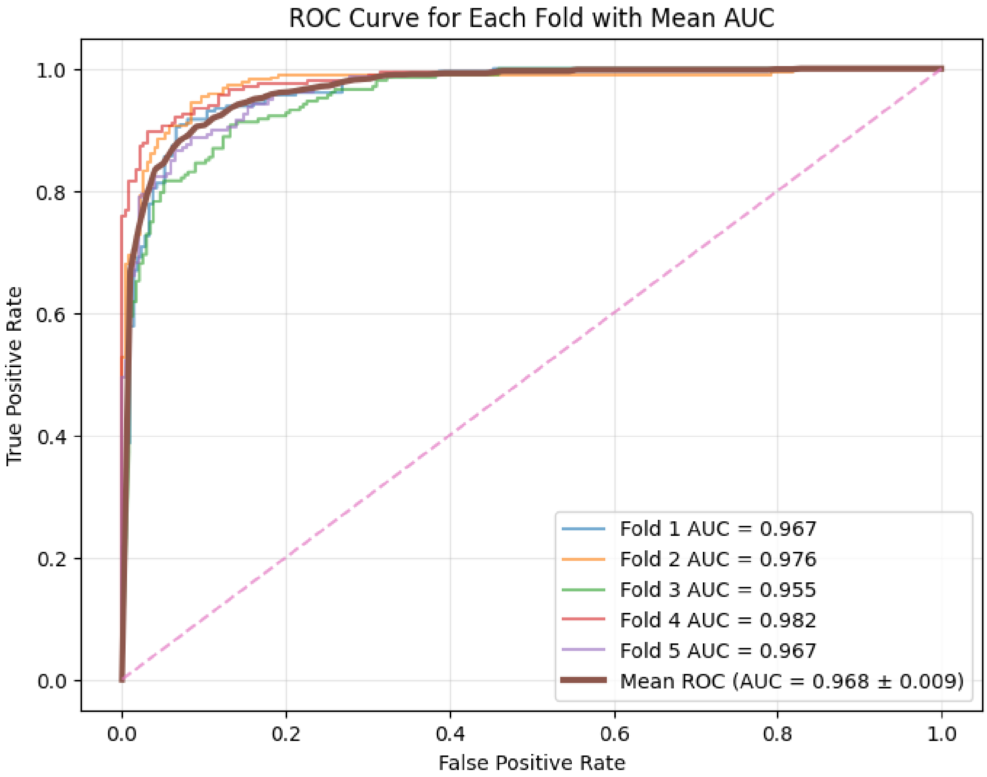


Figure 4.6: The ROC curves of five Naive Bayes models evaluated via 5-fold cross validation

4.7 Decision Tree

The Decision Tree classifier is a technique used that recursively splits the dataset into smaller and smaller subsets. At each split, criteria determine which subset an observation belong to. The end goal is for each of the subsets to be partitioned based on class. In other words, the terminal nodes should have ”pure” subsets only containing points of one class. This creates a tree with a root node where the first decision is made, from which branches with new nodes are made. The branching process stops once a stopping criteria is reached. This criteria could be reaching a certain depth (number of decisions made from the root node), each subset containing a minimum number of samples, or each subset reaching a certain level of purity. At each terminal node, all samples that meet the node’s criteria are classified as the dominant class.

Node impurity may be measured in by different metrics. One way is to measure classification error rate. This is simply the proportion of training observations at a terminal node who's actual class is not the same as the dominant class of that node. For example, if 10 points, 7 red and 3 blue, are members of a terminal node, the error rate would be 3 out of 10 or 30%. All unseen observations that reach this node's criteria will be classified as red. This metric is simple to understand, but often fails to capture the intricacies of class distribution in more complex datasets.

Another way to measure a node's purity is called Gini Impurity. This metric measures probability that a randomly picked data point will be misclassified according to the class distribution of the the terminal node. This is represented by the equation:

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Where C is the number of classes in the set and $p(i)$ represents the probability of picking a sample of class i from the set. A lower Gini Index represents higher purity.

The last common measure of purity is Entropy. This metric is the measure of uncertainty within a node's class distribution. This metric may be calculated by the equation:

$$H = -\sum_{i=1}^{C} p_i \log_2 p_i$$

Where C is the number of classes in the set and $p(i)$ represents the proportion of samples of class i within the set. Lower entropy represents higher purity. Entropy is more sensitive to small fluctuations in class ratio when compared to simple error rate or Gini Impurity.

Using these metrics aid in choosing optimal decision criteria at each node. Overfitting is commonly seen as a problem in a decision tree's performance, even with optimal decision criteria. This occurs when a tree becomes too deep and becomes too specific to the training data. To avoid overfitting a decision tree, a technique called pruning is used. Such as real tree pruning, this technique removes branches of the decision tree that have little predictive power. In doing so, the tree improves in both generalizability and computational performance.

Overall, decision trees are a useful predictive tool in machine learning. Their versatility, being applicable to both regression and classification problems. The ease of understanding decision trees also makes decision trees a useful tool.

The decision tree classifier that we applied to our data set proved to be a useful predictive tool with the highest accuracy we have seen yet when all the data is used (0.999). The classifier ceases to have the highest accuracy when the data is split into half for testing and training and when we apply five-fold cross-validation as well (0.889 and 0.898 respectively). Specificity is once again higher than sensitivity, with the exception of when all of the data is used, however the two values are very similar (1.0 and 0.998). This indicates that the classifier has an easier time predicting the customers with low income.

When looking at the ROC curve we begin to see the flaws of the decision tree classifier. Firstly, the curves of each fold to begin to get close to the top corner of the graph, but then sharply turn horizontal around TPR = 0.850. We also see that the AUC values have an average of 0.892, which is lower than that of the other classifiers examined thus far.

Decision Tree											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.999	0.889	0.898	1.000	0.888	0.893	0.998	0.890	0.904	0.083	0.025	0.197

Table 4.6: Accuracy, sensitivity, specificity, and run time for Decision Tree using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

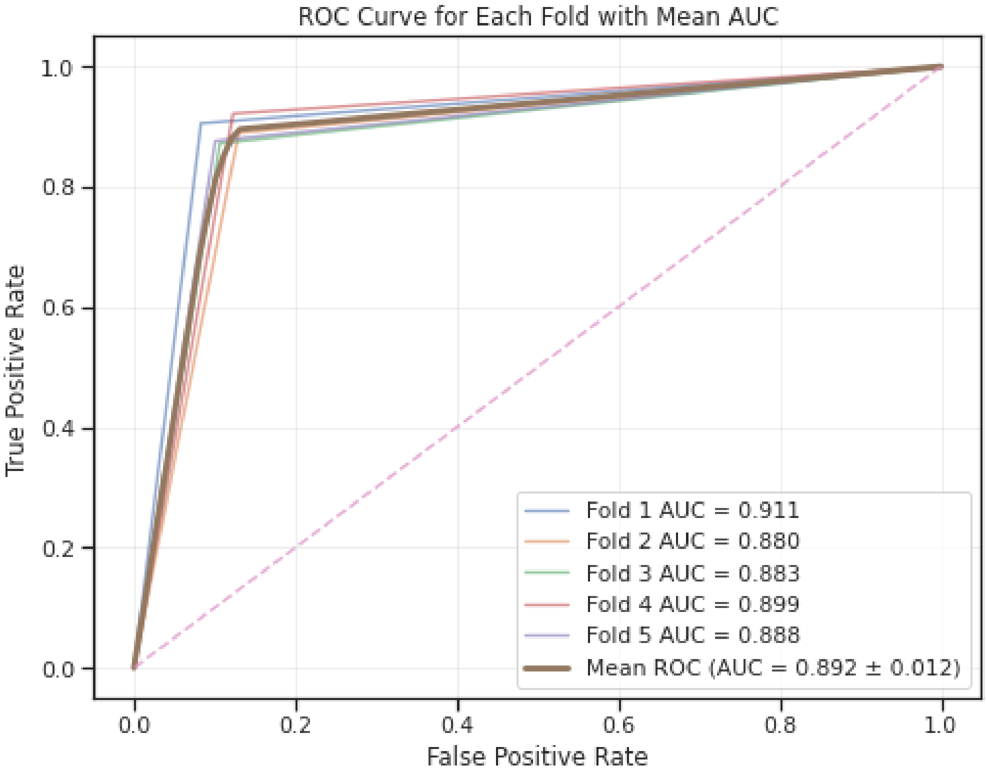


Figure 4.7: The ROC curves of five Decision Tree models evaluated via 5-fold cross validation

4.8 Bagging

Bagging, short for Bootstrap Aggregating, is a machine learning technique that uses multiple versions of a model to produce a model more accurate than its decomposed models alone. This is accomplished by bootstrapping, a technique in which one randomly resamples the sample data with replacement and the same sample size of the original dataset. In doing so, the newly created data sets vary slightly from one to another. Points are repeated in some while missing in others. For each resampled set a new classifier model is created. This process is repeated a predetermined number n times. At the end there exists n classifier models, each trained on a resampled distribution of the original sample. When predicting the class of a data point, the point is given to each tree for which each gives a result. The results are aggregated together to come to a single final prediction.

This technique commonly uses decision trees as the classifier. These models commonly suffer from high variance and overfitting. The random variation of data each model is trained on counteracts the overfitting that occurs in individual models once combined.

Bagging demonstrates strong and stable performance across all evaluation strategies, as shown in Table 4.7. When trained on the full dataset, the model achieves an accuracy of 0.993, with sensitivity 0.966 and specificity 0.990. Performance remains high under the 50% split with an accuracy of 0.891, and under 5-fold cross validation with an accuracy of 0.908, sensitivity of 0.919, and specificity of 0.898. Unlike earlier classifiers that rely on a single decision rule or probability model, Bagging combines the predictions of many resampled decision trees, which reduces the influence of noisy or unstable patterns. This ensemble averaging explains its reliable generalization, achieved with a cross-validation runtime of only 0.323 seconds.

Bagging											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.993	0.891	0.908	0.996	0.909	0.919	0.990	0.876	0.898	0.072	0.113	0.323

Table 4.7: Accuracy, sensitivity, specificity, and run time for Bagging using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

The ROC curves for Bagging, presented in Figure 4.8, further highlight the model's performance. Across all five folds, the curves rise sharply toward the upper-left corner of the plot, which corresponds to high true positive rates and low false positive rates. The mean area under the ROC curve (AUC) of 0.966 indicates excellent discriminative ability, showing that the Bagging classifier reliably separates the two customer groups across a wide range of decision thresholds.Unlike simpler probalisitic or linear models, Bagging benefits from combining slightly different decision trees, allowing it to smooth out unstable predictions from any single tree. The tight clustering of the ROC curves across folds reflects this stabilizing effect and reinforces the reliability of Bagging as a high-performing ensemble for income level classification.

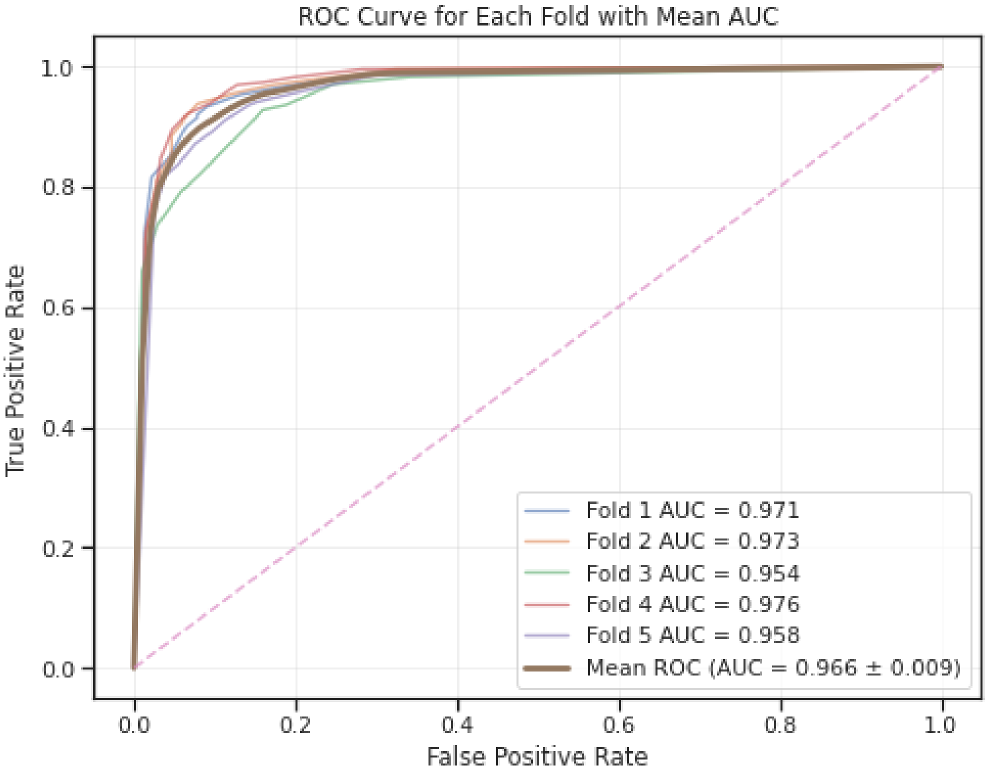


Figure 4.8: The ROC curves of five Bagging models evaluated via 5-fold cross validation

4.9 Random Forest

Random Forest is an ensemble machine learning technique that creates a ”forest” of decision trees, each with randomy selected data and features. Like with Bagging, the response of each tree is then aggregated together to produce a final prediction. Each tree uses a random subset of features, a random subset of training data, and are created independently of each other. The random and independent nature of the trees allows them to all make their own predictions. The differences in prediction helps to reduce overfitting and improves resistance to noise and overall accuracy.

Random Forest can be used for both regression and classification. For regression, the average of the outputs of the trees is often used as the final prediction. In classification, simple majority is the common technique to aggregate the outputs of the trees. The different responses from different learned patterns helps counteract the overfitting that occurs in a single tree. However, random forests lack the transparency and ease of understanding that a single decision tree boasts.

The number of decision trees created is predetermined. Less trees have tend to be susceptable to overfitting and variance, similar to a single decision tree. Important patterns in the data may be missed and overall accuracy tends to be lower the fewer trees are present. The accuracy and consistency of predictions increases with the number of trees, up to a certain point. The computational performance of a forest degrades with more trees and becomes slower to run.

The Random Forest classifier exhibits exceptionally strong performance across all evaluation strategies, as reported in Table 4.8. When trained on the full dataset, the model acheives accuracy of 0.999, with sensitivity 0.998 and specificity 1.000, indicating near-pefect classification in this instance. Under the 50% split, performance remains high with an accuracy of 0.896, sensitiitivty of 0.895, and specificity of 0.896. The 5-fold cross validation results further confirm the model's robustness, yielding an accuracy of 0.907, sensitivity of 0.904, and a specficity of 0.911 These results show that Random Forest not only performs extremely well on see data, but also gernalizes reliability to unseen customer profiles. The cross validation runtime of 1.500 seconds reflects the added computational cost of combining many randomized trees, though the cost remains reasonable relative to the predictive gains.

Random Forest											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.999	0.896	0.907	0.998	0.895	0.904	1.000	0.896	0.911	0.617	0.398	1.500

Table 4.8: Accuracy, sensitivity, specificity, and run time for Random Forest using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

The ROC curve for the Random Forest classifier, shown in Figure 4.9, demonstrate strong and highly consistent classification performance across all five cross-validation folds. The individual fold AUC values range from 0.967 to 0.982, with a mean AUC of 0.972 ± 0.006. This narrow spread of the curves near the top left corner of the plot reflects high true positive rates with low false positive rates over a wide range of thresholds. Together, these results confirm that Random Forest provides excellent overall separation between the two customer spending groups and maintains reliable performance under cross validation.

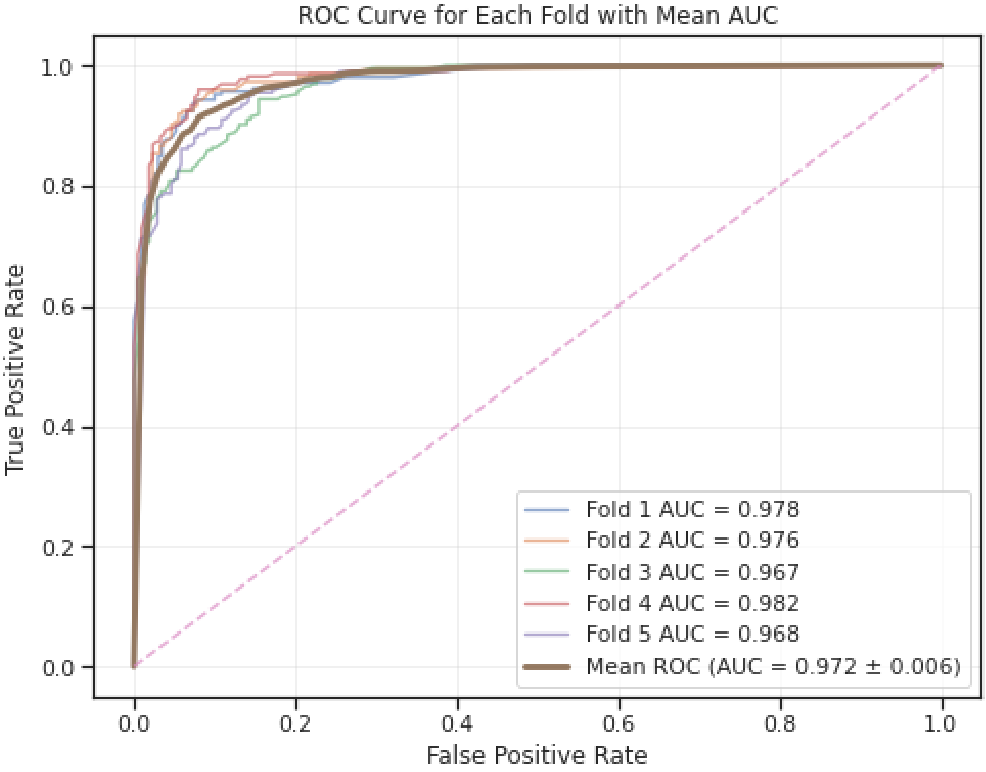


Figure 4.9: The ROC curves of five Random Forest models evaluated via 5-fold cross validation

4.10 AdaBoost

AdaBoost, or Adaptive Boosting, is a machine learning algorithm that uses multiple weak classifiers to create a strong classifier. This algorithm is not a classifier on its own, but it adapts other weak classifier algorithms to ”boost” their performance in conjunction with other classifiers. A popular weak classifier is a decision stump: a decision tree with a depth of 1. To booth these classifiers' performance, the algorithm starts with an initial model with a base weight value assigned to each data point. This model may correctly classify some observations but not others. The correctly classified results have their weights decreased while the wrongly classified observations' weights are increased. These weights are then passed on to another classifier. This time the model emphasizes correctly identifying points with higher weight. The observations again have their weights recalculated according to accuracy and passed to the next model. The process repeats for a predefined number of models. At the end, the weighted predictions of each model are totaled and signed, resulting in a binary outcome of -1 or +1.

This method is an ensemble learning technique, similar to random forest. It aggregates the results of multiple models to give a single response. However, unlike random forest, AdaBoost is builds models sequentially. Each individual model is dependent on the results of the previous. This helps each model "learn" from the mistakes of the last, leading to reduced bias and variance and increased generalizability.

ADA Boost											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.924	0.909	0.918	0.926	0.899	0.920	0.922	0.921	0.917	5.085	7.040	16.82

Table 4.9: Accuracy, sensitivity, specificity, and run time for ADA Boost using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

As seen in Table 4.9, the ADA boost classifier shows a very strong performance in terms of accuracies (0.924, 0.909, 0.918 respectively). This is likely because of the self-teaching nature of the ADA boost classifier, as it learns from past mistakes in previous iterations. The specificity values of of Table 4.9 are higher than that of its sensitivity values, indicating that this classifier will be partial to selecting low income as its prediction.

The ROC curve carries on the idea of the ADA boost being a sufficient predictor of Income. This can be seen from Figure 4.10, where each of the curves converge nearly perfectly into the top left corner of the graph. Additionally, all of the AUC terms exceed 0.970, with an average AUC term of about 0.976. These findings, along with the high accuracies of the ADA boost classifier, prove that ADA boost is a sound predictor of Income.

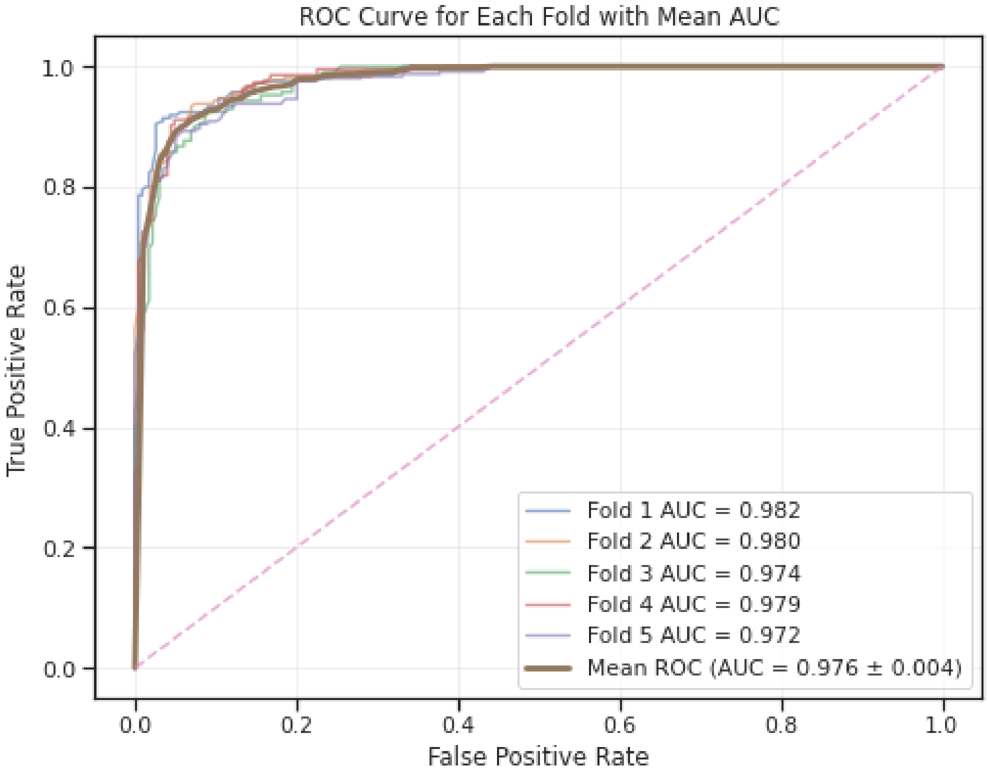


Figure 4.10: The ROC curves of five AdaBoosting Tree models evaluated via 5-fold cross validation

4.11 XGBoost

Extreme Gradient Boosting (XGBoost), is an advanced ensemble learning method built upon the concept of boosting. Unlike Bagging which trains the model independently, boosting trains the models sequentially, where each new model focuses on correcting the mistakes made by the previous ones.

The XGBoost continues to display some of the most impressive data seen from the classification process. We see accuracies of 0.981, 0.909, 0.918 respecitvely, which is the highest across the the three ways we used our data. It is also important to note here that in all but the cross-validation method, sensitivity is higher than specificity, meaning that this classifier will tend to predict higher income more often.

XGB Boost											
Accuracy			**Sensitivity**			**Specificity**			**Run Time (s)**		
Full	50%	CV	Full	50%	CV	Full	50%	CV	Full	50%	CV
0.981	0.909	0.918	0.986	0.941	0.914	0.975	0.882	0.926	0.229	0.120	0.929

Table 4.10: Accuracy, sensitivity, specificity, and run time for XG Boost using full data for training/testing, 50% data for training/testing, and 5-fold cross validation

The ROC curve per Figure 4.11, continues to reinforce why this classifier performed with the best of the classifiers we used. The curves for each fold converge to the top left corner of the graph greatly, indicating a strong and consistent performance. The average AUC for this graph is 0.976, which is very high, yet again reiterating that the XGBoost is one of the best classifiers for predicting Income.

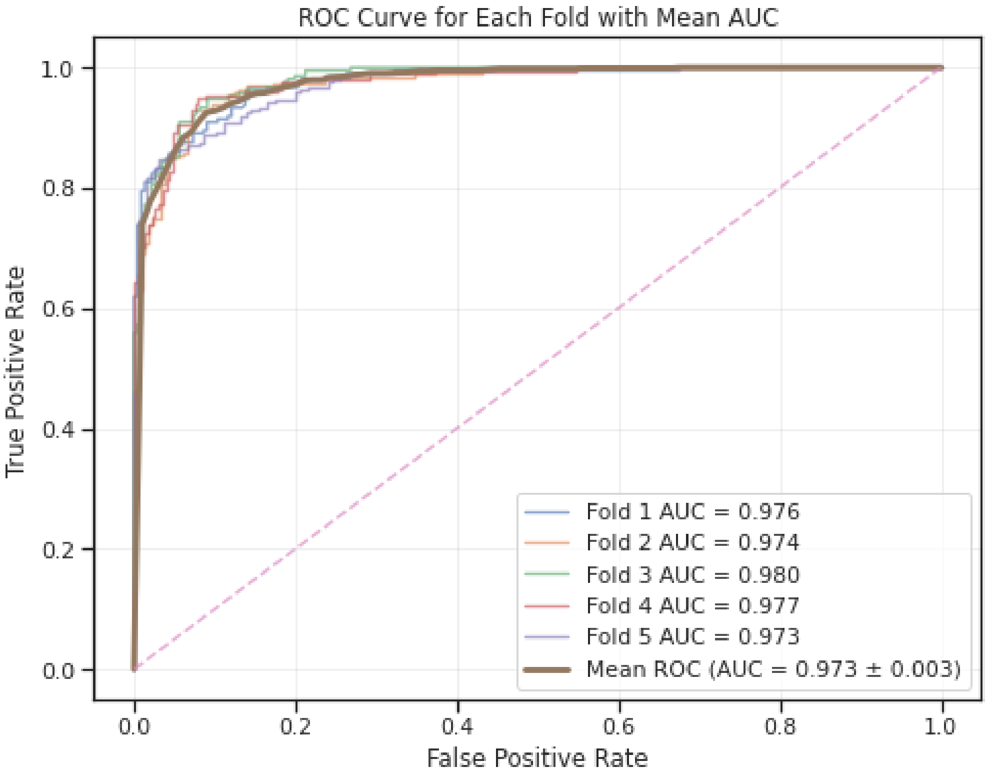


Figure 4.11: The ROC curves of five XG Boost Tree models evaluated via 5-fold cross validation

4.12 Evaluating Models

Summaries of the mean classification model accuracies are plotted in figure 4.12. From this bar graph we can conclude that XGBoost has the highest mean accuracy of each run of the model at 92%. Figure 4.13 confirms that XGBoost has the highest max accuracy as well as highest median accuracy. From figure 4.13 we can see a notable outlier in Naive Bayes classifier at 82% accuracy. Naive Bayes also the lowest mean accuracy at 88.3%, yet in the aforementioned figure we can see a median accuracy of almost 90%. KNN's median is the lowest at 88%.

Figure 4.14, shows AdaBoost has the longest runtime, with Full, 50%, and 5-fold CV runs taking longer than 5 seconds. While there is a general trend of 5-fold CV having longer runtimes than Full or 50%, AdaBoost shows all-around poor training performance compared to all other models. Logistic Regression had a 5-fold CV runtime of 27.914 seconds, which stands out as the longest runtime of all models and runs. Decision Tree has the lowest 5-fold CV runtime at 0.197 seconds. This shows the computational performance of the algorithm. Its mean accuracy of 89.5% is mediocre when compared to other models. This demonstrates the tradeoff between accuracy and performance.

4.13 Best Models AdaBoost & XGBoost

Based upon the comparsions made graphically in figure 4.13 and figure 4.12 it is clear that the two best classification methods are ADA boost and XGBoost. This is shown via the high accuracy values shown in the bar charts and boxplots for comparison, where each of them share virtually the same shape which exceeds the other boxplots on the comparison figure. Additionally the ROC curves for each of the classifiers show very similar features too, where each of the curves for each fold converge to the top left corner of the graph and each of the AUC values show very strong performances. Taking those observations into account, it is clear that the ADA boost and XGBoost are the best predictors for Income status. However, when taking into account the figurefig:model-rt-box, we can see that the ADA boost classifier

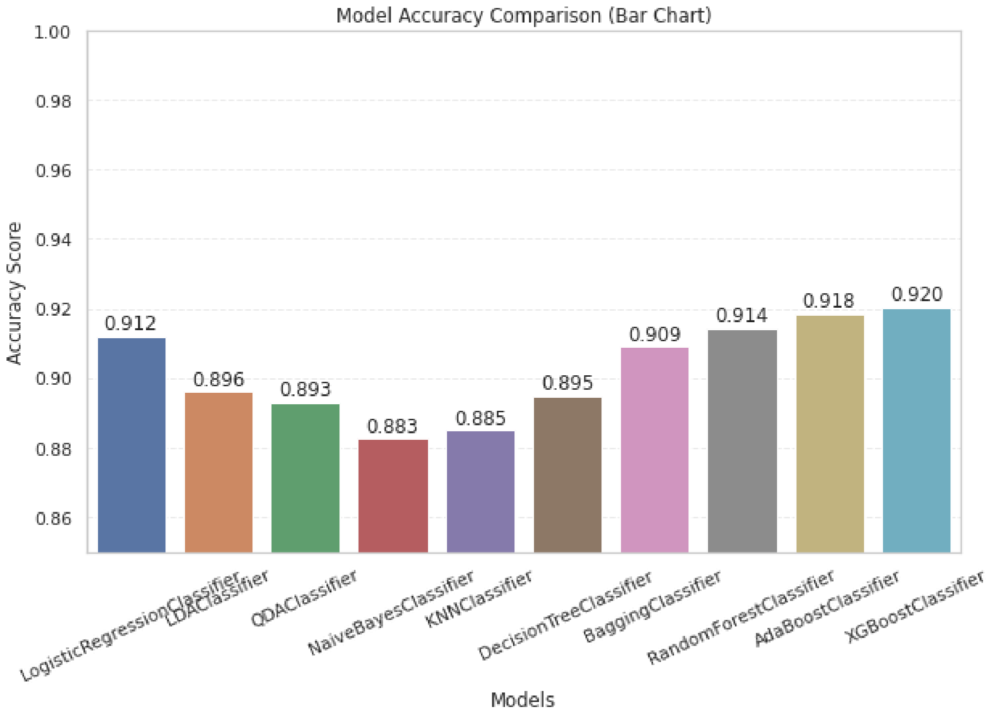


Figure 4.12: Barplot of all model's accuracy

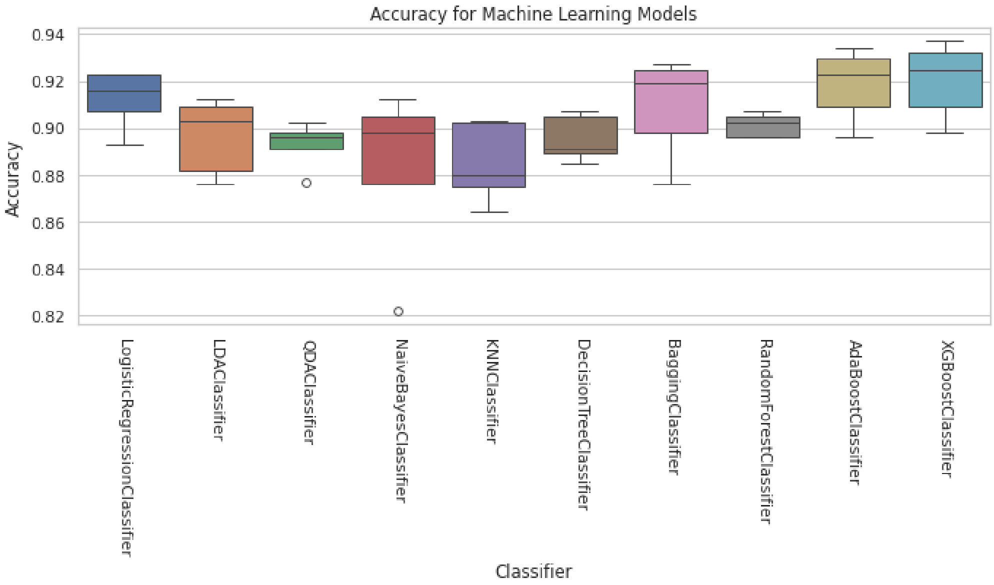


Figure 4.13: Boxplot of all model's accuracy

struggles greatly with it's runtimes, giving the slight edge to the XGBoost classifier.

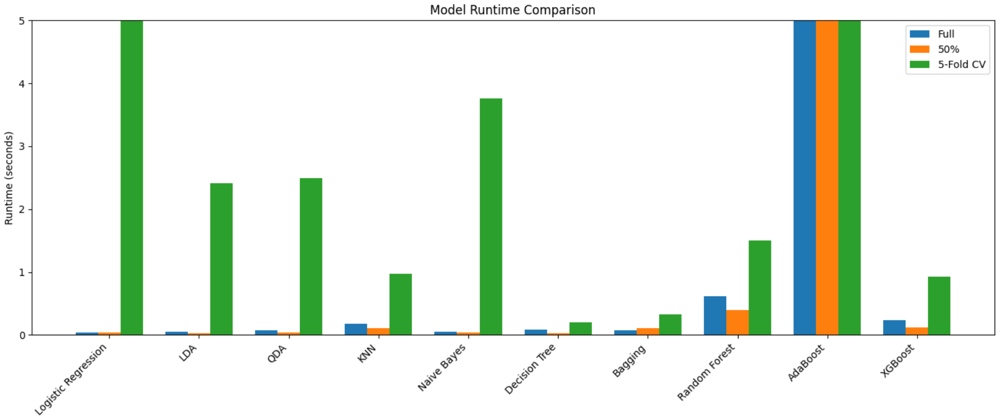


Figure 4.14: Barplot of each model's runtime. Limited to five second due to Logistic Regression and AdaBoost outliers.

5 Conclusion

In this paper, we had two main goals with our consumer profiles dataset. The first was to develop a regression model that can predict consumer income using linear regression. In doing so, we also wanted to know which features had high predictive power and those that did not. The second was to find a classification model that would be able to predict binary categories of consumer income. We successfully found a linear model that was able to predict consumer income within an average of 10,150 euros, using only five feature of the 28 available in the dataset. We found the amount spent on wine and meat, the number of purchases made by catalog, number of store website visits in a month, and number of teens in the home to have the highest predictive power. With an adjusted R^2 of 0.739, this model demonstrated that customer spending patterns paired with basic demographic information can effectively predict income levels while avoiding the complexity of our original 16-feature base model.

Our classification analysis examined ten different methods for distinquishing between high and low income customer segments. The ensemble methods specifically AdaBoost and XGBoost, emerged as the strongest performers, achieving cross validation accuracies exceeding 91% and AUC values above 0.97. These models consistently outperformed traditional approaches like logistic regression and LDA while maintaining reasonable computable efficiency. XGBoost had a particular advantage in runtime performance compared to AdaBoost.

Together, these findings can provide retail businesses with practical tools for tarketing marketing strategies. The regression model reveals which customer behaviors particularly - premium product spending and catalog usage - most strongly indicate higher income levels. Meanwhile our classification models enable businesses to automatically segment their entire customer base at scale, allowing for differentiated campaign strategies such as premium catalogs for high-income customers and deal-focused promotions for low-income customers.

References

[1] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*, Springer, 2023.

[2] imakash3011, *Customer Personality Analysis*, Kaggle, https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis, accessed October 15, 2025.

[3] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, *API design for machine learning software: experiences from the scikit-learn project*, in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122, 2013.