Project Report: A Data-Driven Approach to Customer Segmentation

Introduction

The goal of this project was to address a core challenge for a retail client: how to speak more effectively to different customer groups. The primary objective was to identify and understand two key segments of the population for marketing purposes: individuals who earn less than $50,000 annually and those who earn more. The provided dataset from the U.S. Census Bureau's 1994-1995 surveys, containing 40 demographic and employment-related variables. My role was to build a predictive model and develop a segmentation model that would translate large data into clear, actionable insights for making strategic marketing decisions.

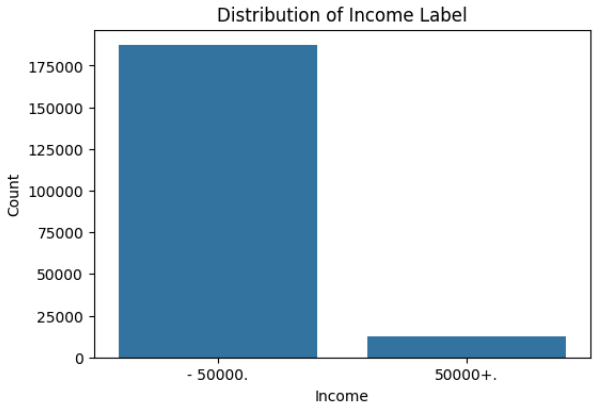
This report shows my process from start to finish. I began with an exploration of the data, such as missing information and a significant imbalance between the two target groups. Following this, I moved into a data cleaning and feature engineering part. Here, I developed a specific technique to convert the most important categorical variables into numerical features by calculating the percentage of high-income earners within each category. This allowed me to retain valuable information while preparing the data for modeling. I also addressed the class imbalance by strategically downsampling the majority group to ensure the model would pay equal attention to both low and high-income individuals.

With a clean, balanced, and well-engineered dataset, I proceeded to the modeling stage. Recognizing that the data spanned two years, I chose to use the 1994 data for training and the 1995 data for testing. This approach simulates a real-world scenario, ensuring the model is evaluated on its ability to predict future outcomes, not just its performance on historical data. I trained and compared three different models: Logistic Regression, XGBoost model, and Decision Tree. While the XGBoost model produced very high accuracy, its complexity made it a "black box," making it difficult to explain the "why" behind its predictions. After careful evaluation, I selected the Decision Tree as the final model since it provides a strong balance between predictive accuracy and interpretability that can be easily understood by non-technical stakeholders and directly inform business strategy.

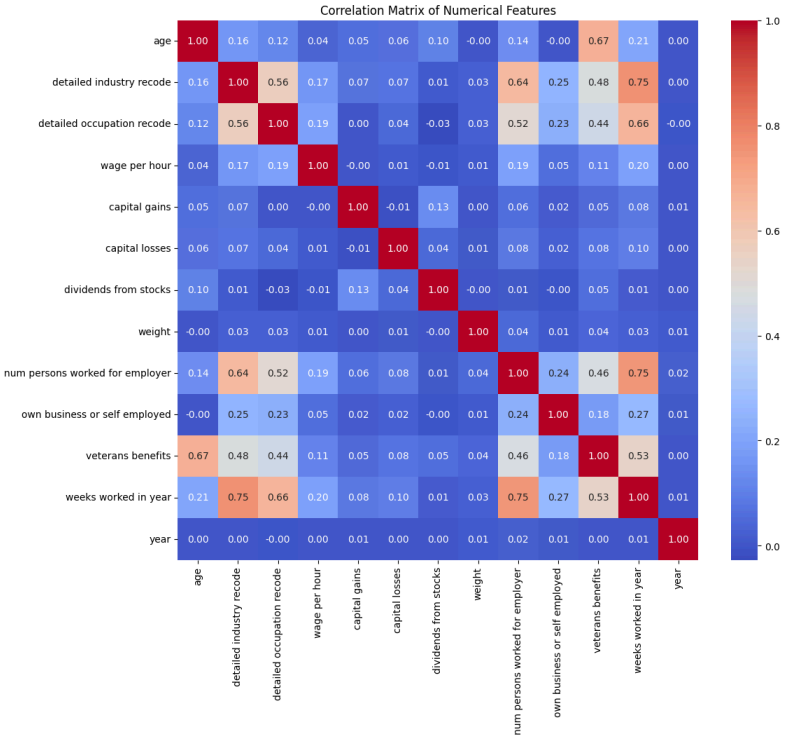
Exploratory Data Analysis (EDA)

Before I could build any predictive models, my first step was to thoroughly explore the dataset. The goal of exploratory data analysis (EDA) is to understand the story the data tells, identify its strengths and weaknesses, and uncover the clues that will later guide the modeling process. My initial inspection confirmed I was working with a substantial dataset of nearly 200,000 individual records, each described by a rich mix of numerical features like age and wage per hour, alongside numerous categorical features such as education and occupation.

The first finding was the distribution of the income (the target variable) is imbalance. The majority of individuals in the dataset earn at or below $50,000, while the higher-income group represents a much smaller fraction of the population. The plot below shows that the dataset is heavily skewed towards the lower-income category. A naive model could achieve over 90% accuracy by simply guessing "less than $50,000" for everyone, but it would completely fail at the primary goal: identifying potential high-income customers. This finding informed me to use specific techniques to ensure the model would learn to recognize the patterns of the smaller, also high-income group.



Then, I found that many categorical columns contained placeholder values like ? and Not in universe, which represented missing or irrelevant data that needed to be cleaned.
Next, I move on to the categorical features like education, marital stat, major occupation code, and sex by visualizing the relationship between different features and income. The data clearly showed that males were significantly more likely to be in the higher-income (>$50k) group than females. It was a strong and consistent pattern across the dataset and served as a logical foundation for a customer segmentation strategy later in the project. Other features also showed clear relationships with income. For example, there was a predictable trend with education, where higher levels of academic achievement corresponded to a greater likelihood of earning over $50,000. Employment status also played a clear role, with individuals in full-time work having much higher earnings potential than those not in the labor force.
Finally, I analyzed the numerical features. While many(age) showed a reasonable distribution, others related to wealth, such as capital gains and dividends from stocks, were highly skewed. This indicated that most people had no income from these sources, while a small minority did.
To understand the interplay between these numerical variables, I generated a correlation matrix.



This heatmap above shows the linear relationships between variables. While some employment-related features are moderately correlated, many, like capital gains and age, provide independent information. The heatmap confirmed that while some employment features were related, there were no overwhelmingly strong correlations that would suggest redundant information.

After EDA, I moved on to data preparation. I cleaned the data and engineered features to transform the original dataset into a high-quality, structured format optimized for machine learning.

Since the dataset contained both standard null values (NaN) and placeholder strings like ?, I replaced all such instances across the categorical columns with a single, uniform value 'Unknown_Category'. Then I converted the target variable, label, from its text format ('- 50000.', '50000+.') into a binary numerical format. I mapped the lower-income group to 0 and the higher-income group to 1, creating the label_encoded column that would serve as the target for all subsequent classification tasks.

Before feature engineering, I split the data. The dataset contains records from two distinct years, 1994 and 1995, therefore, I designated all data from 1994 as my training set and reserved the entire 1995 dataset as my test set. To build a model that can generalize to new, unseen data. After splitting, the year column was no longer needed and was removed from both datasets.

Based on the insights from EDA, I selected a group of the most promising categorical features for transformation. Rather than using one-hot encoding, which would have created a very wide and sparse dataset with hundreds of new columns, I chose target encoding, which can directly capture the relationship between a categorical feature and the target variable in a single numerical value.

After all features were converted to a numerical format, I downsampled the training set to prevent my models from developing a bias towards the majority low-income class.

Finally, I conducted a quick feature assessment to validate my engineering process. I trained a simple Logistic Regression model on the cleaned and balanced dataset. The purpose was to inspect the model's coefficients. These coefficients provide a clear indication of the direction and magnitude of each feature's influence on the probability of earning over $50,000. The results showed the target-encoded features for sex emerged with the largest absolute coefficients, and now the dataset was ready for modeling.

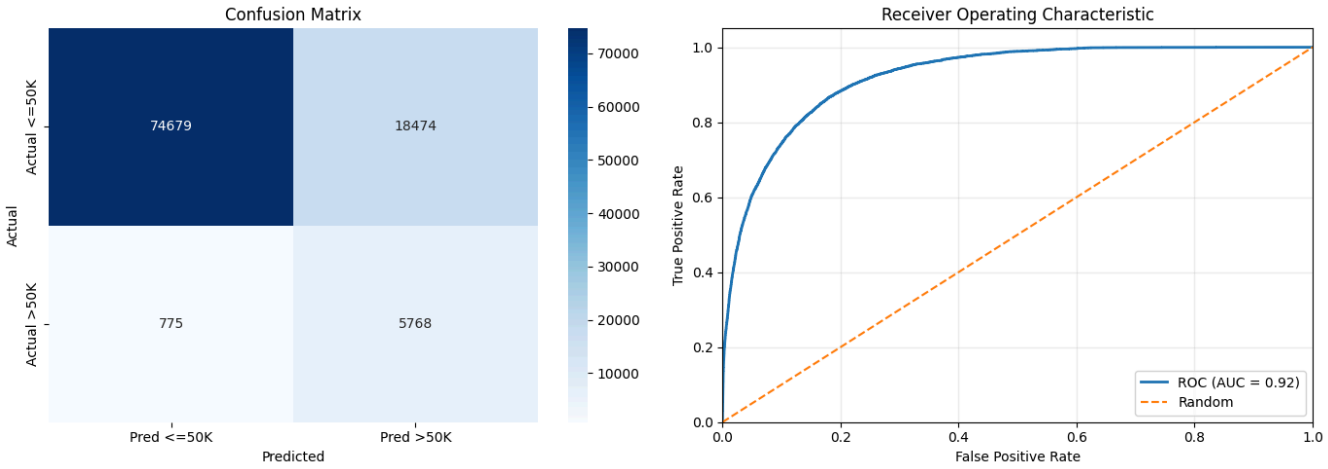
Modeling

With a clean, engineered, and balanced dataset, I moved to the modelling part. My objective was to not only create a model that could accurately classify individuals into the two income brackets but also to select a final model that was transparent, interpretable, and could provide actionable business insights. To achieve this, I trained and compared three distinct models: a Logistic Regression to serve as a strong baseline, a XGBoost model to test the limits of predictive accuracy, and a simple Decision Tree to prioritize clarity and interpretability.

Each was trained on the balanced 1994 dataset and then evaluated on the imbalanced 1995 test dataset to simulate a real-world performance scenario.

Model 1: Logistic Regression

My first step was to build a Logistic Regression model. This is a robust and highly interpretable linear model that estimates the probability of an individual belonging to the high-income class. It provides an excellent baseline to measure the performance of more complex models.
After training the model, I evaluated it on the 1995 test data. The model achieved an overall accuracy of 80.7%. However, for a business problem with imbalanced classes, a deeper look is required. The classification report revealed a critical trade-off. The model achieved a very high recall of 88% for the high-income (>50K) group. The model is extremely effective at finding the vast majority of high-income individuals in the population. This ensures that very few potential high-value customers are missed by our marketing efforts. However, this came at the cost of low precision (24%). This means that when the model predicted an individual was in the high-income group, it was only correct about a quarter of the time. This would lead to inefficiency, as a significant portion of a targeted marketing campaign's budget would be spent on individuals who are not actually in the target segment. The confusion matrix below visually illustrates this, showing that while we correctly identified 5,768 high-income individuals (True Positives), we incorrectly labeled 18,474 low-income individuals as high-income (False Positives).

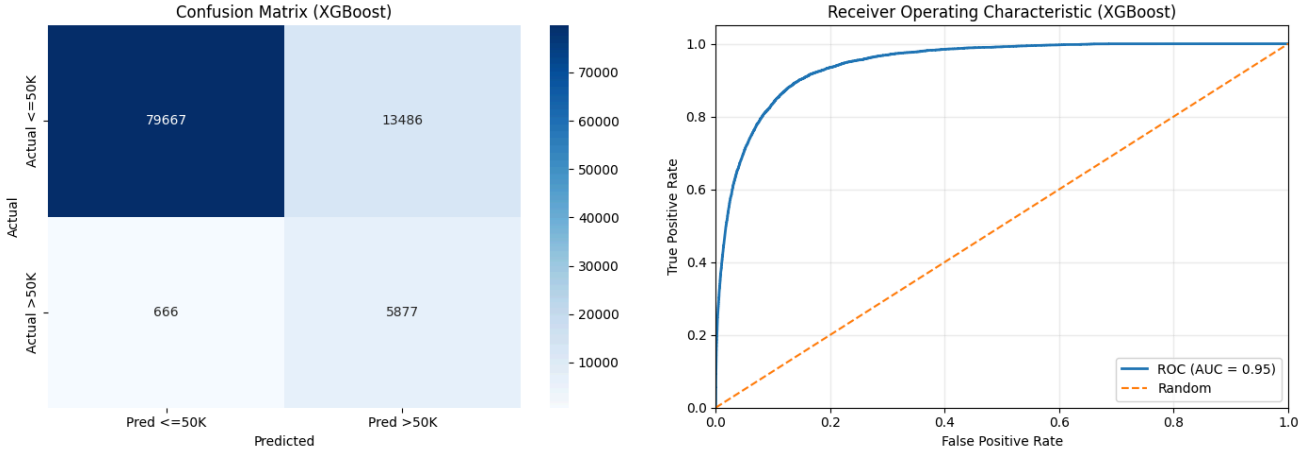


The model correctly identifies most high-income earners (5,768) but also misclassifies a large number of low-income earners (18,474), indicating high recall but low precision.
Finally, the model's overall classification power was measured using the ROC curve and its corresponding AUC score. An AUC of 0.92 demonstrates excellent overall performance, indicating the model is highly effective at distinguishing between the two income classes. The model achieved an AUC of 0.92, which is an excellent score. It indicates that the model has a 92% chance of correctly ranking a random high-income individual higher than a random low-income one. In summary, the Logistic Regression model served as a very strong baseline: powerful at identifying potential targets but inefficient in its precision.

Model 2: XGBoost

Next, I wanted to see if I could improve upon the baseline using a more powerful and complex algorithm. I chose XGBoost, an advanced model that builds hundreds of decision trees

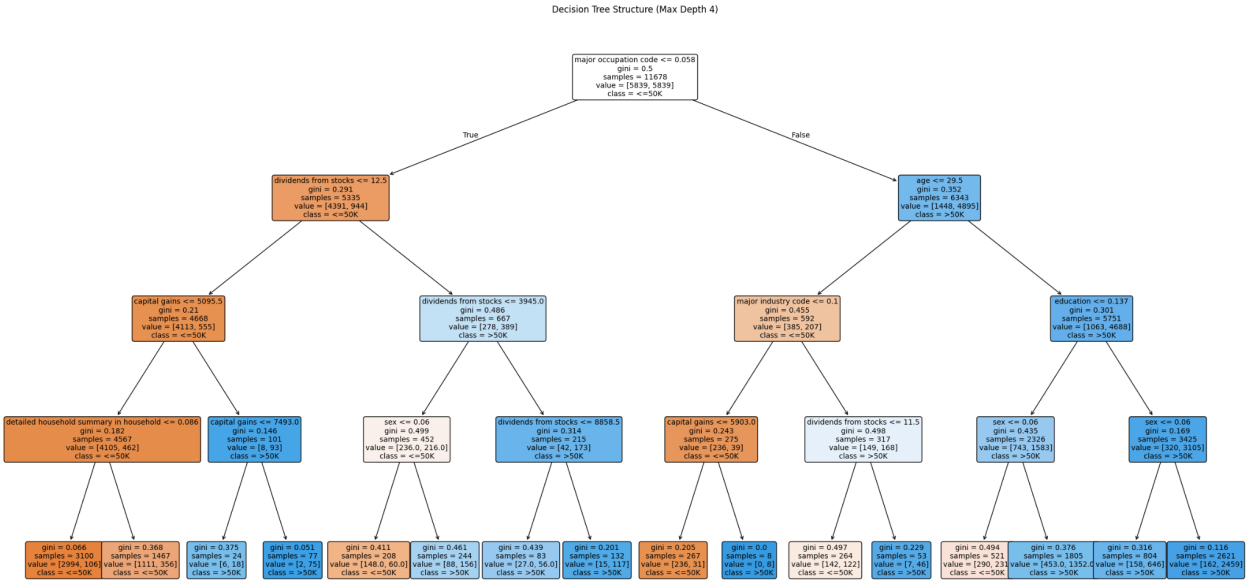
sequentially, with each new tree learning from the mistakes of the previous ones. It is renowned for its state-of-the-art performance. The results immediately showed a significant improvement in predictive power. The XGBoost model achieved a higher overall accuracy of 85.8%. More importantly, it improved on the key business metrics. Recall for the high-income group increased to an outstanding 90%, meaning it found even more of our target customers. At the same time, precision saw a meaningful jump to 30%. This represents a considerable improvement in marketing efficiency, as it would reduce the number of incorrectly targeted individuals.



XGBoost improves on the baseline, increasing True Positives to 5,877 and significantly reducing False Positives to 13,486. The model's performance is better, which is near-perfect and indicates an exceptional ability to separate the two classes. With an AUC of 0.95, the XGBoost model shows the highest level of predictive accuracy among the models tested. Right now, the XGBoost model was the clear winner. However, its power comes with a significant business drawback: it operates as a "black box." Due to its complexity, it is extremely difficult to understand the exact reasons behind a specific prediction. I could tell the marketing team who to target, but I could not easily explain why. For building a sustainable and understandable strategy, this lack of transparency is a major limitation.

Model 3: Decision Tree

My final model was Decision Tree. I intentionally constrained its complexity by limiting its maximum depth to four levels. This was a deliberate strategic choice to prioritize interpretability over raw predictive power. The goal was to see if I could achieve competitive performance while gaining complete transparency into the model's logic. The Decision Tree achieved an accuracy of 81.0% and an AUC of 0.92, placing its overall performance on par with the strong Logistic Regression baseline. While its precision and recall metrics were slightly different, it offered a compelling balance.The true value of this model, however, is not in its metrics but in its structure. Unlike the other models, the Decision Tree provides a clear, intuitive flowchart that explains exactly how it arrives at a decision.



This visual representation of the model provides clear, rule-based logic. Each node represents a decision based on a feature, making the model's predictions fully transparent and explainable. We can read this tree like a set of business rules. For instance, the very first split is on major occupation codes. If an individual's score for this feature is low (indicating a less specialized or lower-paying profession), the model then looks at their dividends from stocks. If they have any dividend income, their
chances of being in the high-income group increase dramatically. This is not just a prediction; it's a powerful business insight. It tells us that for certain population segments, even a small amount of investment income is a major differentiator. These are the kinds of actionable insights that can directly inform marketing strategy.

When comparing the three models, my choice was based on the project's business goals. Logistic Regression is a solid baseline with good recall but poor precision. XGBoost is the top performer in terms of raw accuracy and efficiency, but a "black box" that is difficult to interpret. Decision Tree shows competitive performance on par with the baseline, but with the invaluable benefit of complete transparency. While XGBoost offered the best numbers, its inability to explain its reasoning makes it a less strategic tool. A model that cannot be understood cannot be trusted or used to build long-term business intelligence. For these reasons, I recommend the Decision Tree as the final production model. It provides a robust and reliable level of predictive accuracy while offering clear, understandable, and actionable rules. It empowers the marketing team not just with a list of targets, but with the knowledge of why they are targets. This moves the project beyond simple prediction and delivers a true strategic asset that can be used to build smarter, more informed, and more effective marketing campaigns.

Segmentation Model

The final step of the project was to turn the model's predictions into a real-world marketing strategy. A predictive model is useful, but a clear segmentation plan is what makes it actionable. I explored two different ways to create these customer segments.

First, I tested an idea: would a model trained only on male data be better at predicting outcomes for men than a general model? And the same for women? It seemed logical that specialized models might be more accurate. To check this, I split my data by gender and built separate Logistic Regression models for each. I then compared their performance to the main model that was trained on everyone. The result was clear—the specialized models didn't perform any better. In fact, their AUC scores were slightly lower (0.920 for males and 0.921 for females, compared to 0.928 for the general model). This told me that creating separate models was just extra work for no real gain. My main model was already using gender effectively as a feature, so I decided to drop this approach and find a more direct method.

My second approach was much simpler and proved to be far more effective. Instead of building new models, I used the scores from my best-performing model, the XGBoost classifier. I used it to give every person in the 1995 test set a score from 0 to 1, representing their probability of being a high-income earner. From there, the plan was simple: I set a high threshold of 0.80. Anyone with a score above that was flagged as a prime marketing target. I then split this high-score group by gender, creating two key segments: 'High Score Male' and 'High Score Female'. Everyone else fell into an 'Others' bucket.

The results of this strategy were significant. The 'High Score Male' segment was small—only about 8,000 people—but their actual high-income rate was 50.1%. Compared to the 6.6% rate in the general population, this is a 7.6 times improvement, or a "lift" of 7.6x. The 'High Score Female' segment also showed a huge improvement, with a high-income rate of 34.6%, representing a 5.3x lift. Meanwhile, the 'Others' group, which was the vast majority of people, had an extremely low high-income rate of just 1.9%. This proved that the strategy works incredibly well at concentrating marketing efforts on the right people.

So, my final recommendation is to put this score-based plan into action. The marketing team can use the XGBoost model to score new customers. Anyone scoring over 0.80 should be treated as a high-priority target. Within that group, the 'High Score Male' segment should be the top priority for any campaigns involving premium or high-value products, with the 'High Score Female' segment as a strong secondary focus. By concentrating the budget on these small, high-potential groups and spending less on the large 'Others' group, the return on marketing investment should increase substantially. This same scoring method could even be used with other features in the future, like occupation, to find even more specific customer segments.

Conclusion

My goal for this project was to take a large census dataset and build a model for a retail client to better target high-income customers. After going through the entire process, exploring the data, feature engineering, and testing several models, I've developed a complete strategy that successfully pinpoints these valuable individuals.

The most important decision I made came during the modeling stage. While the powerful XGBoost model gave the highest accuracy on paper, it was a "black box," and I couldn't easily explain its logic. For that reason, I chose the simpler Decision Tree as my recommended model. Its performance was still very strong and competitive, but its real advantage is that it is completely transparent. It provides simple rules that show exactly why someone is likely to be a high-income earner, which is something the marketing team can actually understand and use.

The final strategy I'm recommending is simple and highly effective. We use the best-performing model to score all customers on their likelihood of being high-income, and then concentrate on the small group of people who score above a high threshold. My analysis showed that this method is extremely good at finding the right people. The 'High Score Male' segment I identified had a high-income rate 7.6 times higher than the general population, and the 'High Score Female' segment was 5.3 times higher. This is the key result that proves the value of the approach.

What this all means is that the client can now move from broad marketing to a targeted, data-driven strategy. They have a clear method for focusing their budget on the small group of customers with the highest potential, which should lead to more efficient spending and better results from their campaigns.