Income Prediction and Customer Segmentation Report

1. Data Exploration and Understanding

1.0 Data Understanding

The raw dataset consisted of a large and comprehensive collection of records, comprising approximately **43 columns and close to 200,000 rows**. Each column captured distinct demographic, occupational, and economic attributes of individuals. Given the scale and richness of the data, it was essential to develop a thorough understanding of its structure and content before applying any preprocessing or modeling techniques.

Initial inspection revealed that **a majority of the features were categorical in nature**, representing socioeconomic and employment-related characteristics. As most machine learning algorithms require numerical inputs, an early consideration was how these categorical variables could later be transformed into a form that preserves their informational value while remaining computationally effective.

Despite the high dimensionality, the dataset contained relatively few columns that provided vague or non-informative signals. Most features carried meaningful context that could contribute toward distinguishing income levels. As a result, careful judgment was required to determine **which features should be retained and which could be safely removed**, ensuring that potentially predictive information was not discarded prematurely.

During this phase, particular attention was paid to columns containing ambiguous or placeholder values. Rather than immediately removing such features, their structure and relevance were analyzed to determine whether they could serve a purpose during training. Specifically, certain columns were identified as useful for **assigning sample weights**, allowing the model to better account for population representation during both training and evaluation.

A critical observation during this phase was the significant class imbalance in the target variable. Of the approximately 200,000 records, 93.8% (187,141) represented individuals earning $50,000 or less, while only 6.2% (12,382) earned above $50,000. This imbalance had direct implications for model selection and evaluation relying solely on accuracy would be misleading, as a model that predicts every individual as earning below $50,000 would achieve 93.8% accuracy while being entirely useless. This finding informed the decision to prioritize metrics such as ROC-AUC, precision, recall, and F1-score during model evaluation, and to employ techniques like class weighting during training.

To guide downstream decisions, an important objective of this phase was to establish a framework for **comparing feature behavior relative to the target variable, income**. Understanding how different attributes varied across income brackets helped prioritize features for further analysis and informed the direction of subsequent exploratory and preprocessing steps. This foundational understanding of the data directly shaped the exploration, preprocessing, and modeling strategies described in the following sections.

1.1 Data Loading and Initial Inspection

The dataset was successfully loaded and inspected using standard exploratory techniques, including `.info()` and `.describe()`, to understand feature types, distributions, and missing values. This initial inspection revealed that several columns contained inconsistent data types and a high proportion of placeholder values.

One notable issue identified was the **`age` feature**, which was incorrectly parsed as a string. This caused valid values to be treated as nulls during aggregation. The column was explicitly converted to a numeric data type to ensure accurate statistical analysis and downstream modeling.

1.2 Missing and Invalid Values Analysis

A significant number of columns contained placeholder values such as **"Not in Universe (NIU)"** and **"?"**, which dominated certain features. To quantify their impact, we visualized the proportion of these values per column and evaluated whether they exceeded thresholds such as **50% or 90%**.

Before removing any columns, their **potential relevance to the target variable (`income`)** was assessed to avoid discarding features that might still hold predictive value. Columns overwhelmingly dominated by NIU or unknown values and offering minimal interpretability were marked for removal.

---

2. Data Preprocessing

2.1 Feature Selection and Column Dropping

During preprocessing, the following actions were taken:

- Dropped columns with excessive NIU or unknown values.
- Removed features with limited relevance or weak correlation with income.
- Retained categorical features with strong domain significance despite high cardinality.

This decision-making was guided not only by statistical properties but also by **business relevance**, ensuring that meaningful socioeconomic indicators were preserved.

2.2 Categorical Feature Analysis

Exploratory analysis revealed strong income-based patterns across several categorical variables. For example:

- **Major industry and occupation codes** showed clear income stratification.
- Occupations in **armed forces, mining, and manufacturing** tended to earn more than those in **agriculture and retail trade**.

Despite requiring transformation, these features were retained because they provide valuable signals about earning potential.

2.3 Encoding and Transformation

For the classification task, label encoding was used to convert categorical variables into numerical form. This approach was appropriate because XGBoost, being a tree-based model, splits data at individual values rather than assuming ordinal relationships between encoded numbers. For example, a value of 15 is not treated as "greater" than 1 — the model evaluates each split point independently.

For the segmentation task, the encoding strategy was changed to one-hot encoding. K-Means relies on Euclidean distance to form clusters, meaning it would incorrectly interpret label-encoded values as having magnitude and ordering. One-hot encoding ensures each category is represented as an independent binary feature, allowing K-Means to measure distances meaningfully.

---

3. Model Development and Training

3.1 Feature, Target, and Sample Weights

The dataset represents individual-level human data and includes **sample weights**, which were incorporated during both training and evaluation to ensure representative performance estimates.

- **Target Variable**: Income (≤$50K vs >$50K)

- **Features**: Selected numerical and encoded categorical variables
- **Weights**: Applied to account for population representation

3.2 Baseline Model: Logistic Regression

Logistic Regression was used as a baseline model. Given the **class imbalance**, evaluation relied on more than accuracy, including:

- Precision
- Recall
- False negatives
- Confusion matrix
- ROC-AUC

After regularization, the model achieved an **ROC-AUC of ~0.92**, establishing a strong baseline while maintaining interpretability.

3.3 Advanced Model: XGBoost

XGBoost was selected due to its ability to:

- Handle non-linear relationships
- Manage mixed feature types
- Address class imbalance effectively

The `scale_pos_weight` parameter was tuned to emphasize the minority class. This resulted in an improved **ROC-AUC of ~0.95**.

While false positives increased relative to logistic regression, this trade-off was deemed acceptable for the business context. Sending marketing materials to some lower-income individuals is significantly less costly than missing high-income prospects (false negatives), which were substantially reduced.

3.4 Hyperparameter Optimization

Optuna was employed for hyperparameter tuning to explore potential performance gains. This resulted in a modest but meaningful improvement, increasing ROC-AUC from **0.952 to 0.956**. Given the marginal gain, the tuned model was accepted as the final classifier.

3.5 Feature Importance

Understanding which features most influence the model's predictions is essential for providing clarity to the client. Feature importance analysis reveals that the top three drivers are: weeks worked in year, sex, and dividends from stocks.

Weeks worked in year ranks highest, which is intuitive — the more weeks an individual works in a year, the more likely they are to earn $50,000 or more. Sex ranks second, reflecting the significant income disparity present in the 1994-95 census data, where 10.2% of males earned above $50,000 compared to just 2.6% of females. It is important to note that this reflects historical patterns in the data rather than a recommendation for differentiated marketing by gender. Dividends from stocks ranks third, as individuals receiving dividend income are typically investors with higher overall wealth, making them more likely to fall into the higher income bracket.

While education ranked eighth in feature importance, this does not diminish its predictive value. During exploratory analysis, over 54% of individuals with professional school degrees earned above $50,000. However, XGBoost measures importance by how frequently a feature is used for splits, and much of the signal captured by education is already accounted for by features like weeks worked and occupation.

4. Customer Segmentation Analysis

4.1 Motivation for Segmentation

To support actionable business insights, clustering was performed to understand **distinct population segments**. Features were separated into **numerical and categorical groups** to accommodate the distance-based nature of K-Means.

Unlike tree-based models, K-Means is sensitive to scale and magnitude, making proper feature preparation essential.

4.2 Feature Encoding and Scaling

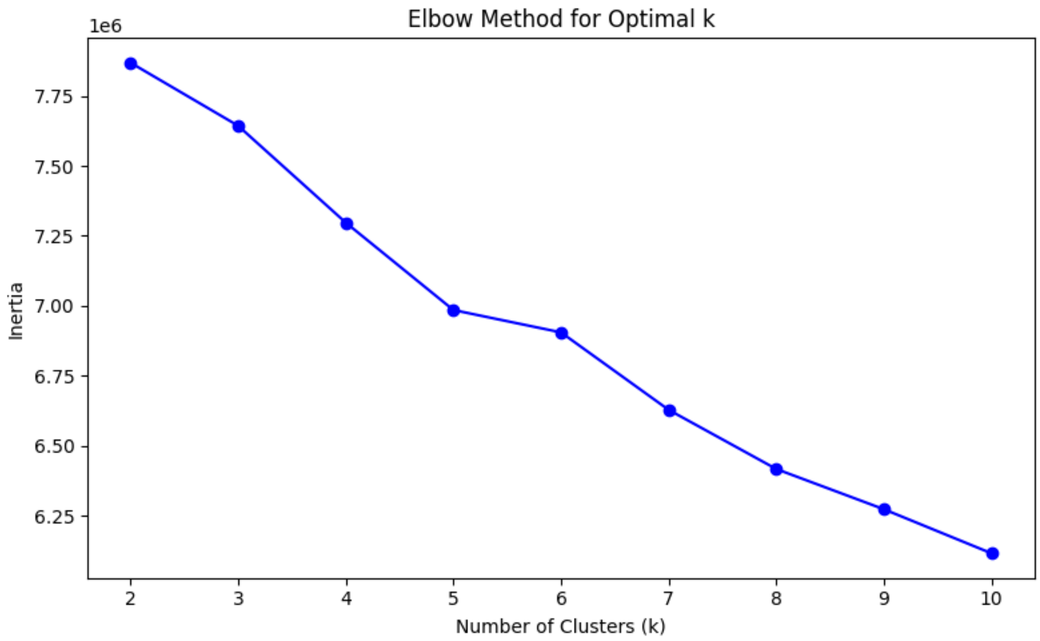
- Categorical features were one-hot encoded.

- Numerical features were standardized to ensure uniform contribution.
- Encoded features were concatenated to form the final clustering input.

4.3 Determining Optimal Clusters

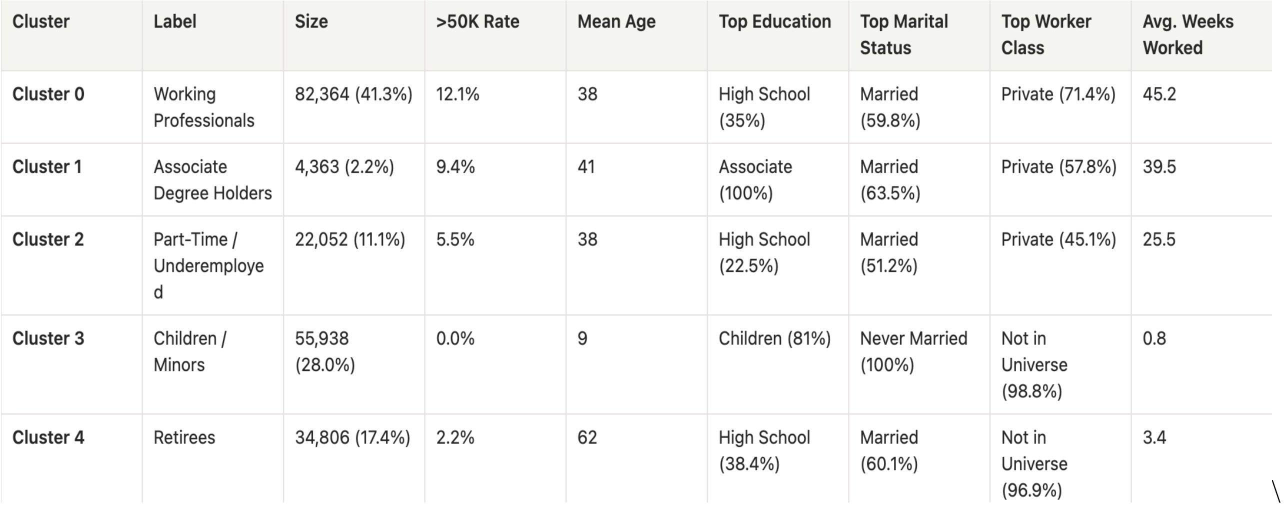
The optimal number of clusters was determined using:

- **Elbow Method**



This method indicated that **five clusters** offered the best balance between interpretability and representational power.

4.4 Clustering Results



Cluster	Label	Size	>50K Rate	Mean Age	Top Education	Top Marital Status	Top Worker Class	Avg. Weeks Worked
Cluster 0	Working Professionals	82,364 (41.3%)	12.1%	38	High School (35%)	Married (59.8%)	Private (71.4%)	45.2
Cluster 1	Associate Degree Holders	4,363 (2.2%)	9.4%	41	Associate (100%)	Married (63.5%)	Private (57.8%)	39.5
Cluster 2	Part-Time / Underemployed	22,052 (11.1%)	5.5%	38	High School (22.5%)	Married (51.2%)	Private (45.1%)	25.5
Cluster 3	Children / Minors	55,938 (28.0%)	0.0%	9	Children (81%)	Never Married (100%)	Not in Universe (98.8%)	0.8
Cluster 4	Retirees	34,806 (17.4%)	2.2%	62	High School (38.4%)	Married (60.1%)	Not in Universe (96.9%)	3.4

\

Fig 1.1 Overall Top Choices

4.5 Cluster Interpretation and Insights

Cluster 0 (Largest Segment – 41.3%)

- Mean age: ~38–40
- Majority married, high school graduates
- Private-sector workers
- Worked >75% of the year
- Higher capital gains (~$811) and wage/hour (>100)
- **12.1% earn >$50K**

This group represents stable, working-age individuals with consistent employment.

Cluster 1 (Smallest Segment)

- Mean age: ~41
- Female-dominated
- Married
- Worked 50–75% of the year
- Lower capital gains (~$400)

- **9.4% earn >$50K**

Reduced work duration appears to correlate with lower income probability.

Clusters 3 & 4 (Children and Retired)

- Minimal or no weeks worked
- Ages skew toward dependents and retirees
- High school education but limited labor participation
- **0% and 2.2% earn >$50K**, respectively

These clusters confirm that age and employment status are dominant drivers of income outcomes.

---

5. Business Judgment and Recommendations

5.1 Model Usage Recommendation

- Use **XGBoost** for income prediction due to superior recall and ROC-AUC.
- Accept higher false positives to minimize missed high-income individuals.
- Deploy the model as a **lead qualification or targeting filter**, not a hard decision rule.

5.2 Marketing Strategy Insights

Marketing efforts should prioritize individuals who:

- Are aged 35–45
- Work consistently throughout the year
- Show higher capital gains or wage/hour
- Belong to stable employment classes

By combining income prediction with cluster membership, the business can **personalize outreach strategies**, improve conversion rates, and allocate marketing spend more efficiently.

Future Directions

This project can be extended beyond offline modeling by deploying the trained income prediction model as an interactive application. One potential direction is to develop a **FastAPI-based backend** with a lightweight frontend such as **Streamlit**, enabling real-time prediction by allowing users to input individual demographic and employment information. This would provide an intuitive way to visualize model behavior while improving accessibility for both technical and non-technical stakeholders.

Another meaningful extension would involve incorporating **client-specific marketing context** into the modeling framework. By explicitly defining the type of product or service being marketed, the model outputs and evaluation metrics could be interpreted in a more targeted manner. This would allow predictions and performance scores to be aligned with concrete business objectives, making recommendations more actionable and relevant.

Several analytical questions also emerged during this work and warrant further investigation. For example, the rationale behind selecting specific income thresholds and time periods could be revisited. Exploring alternative thresholds, incorporating multiple years of data, or extending the analysis across different decades may provide deeper insights into income dynamics and model generalizability. Finally, extensive in-code documentation has been added to ensure transparency and interpretability. All key modeling and preprocessing decisions are clearly explained so that both technical practitioners and business users can understand not only the results, but also the reasoning behind each step. This foundation supports future enhancement, collaboration, and deployment of the system.

References:

- Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011

- Chen & Guestrin, "XGBoost: A Scalable Tree Boosting System," KDD, 2016

- Akiba et al., "Optuna: A Next-generation Hyperparameter Optimization Framework," KDD, 2019

- Scikit-learn documentation — K-Means Clustering, Label Encoding, StandardScaler: https://scikit-learn.org/stable/

- XGBoost documentation — scale_pos_weight for class imbalance: https://xgboost.readthedocs.io/