Income Prediction and Customer Segmentation Analysis

Yihe Huang
yihe@psu.edu

1. Executive Summary

The objective of this project was twofold:

a) Income Prediction - predict whether an individual earns more than $50k annually.

b) Customer Segmentation - Identify distinct demographic groups within the population to inform targeted marketing strategies.

For the classification task, four models were evaluated: Random Forest, XGBoost, Logistic Regression, and a Neural Network (MLP). Their performance was measured using precision, recall, F1-score, accuracy, and ROC-AUC.

To address the class imbalance (only ~6% of individuals earn more than $50k), Conditional Tabular GAN (CTGAN) was used to generate synthetic samples, which improved model performance on the minority class.

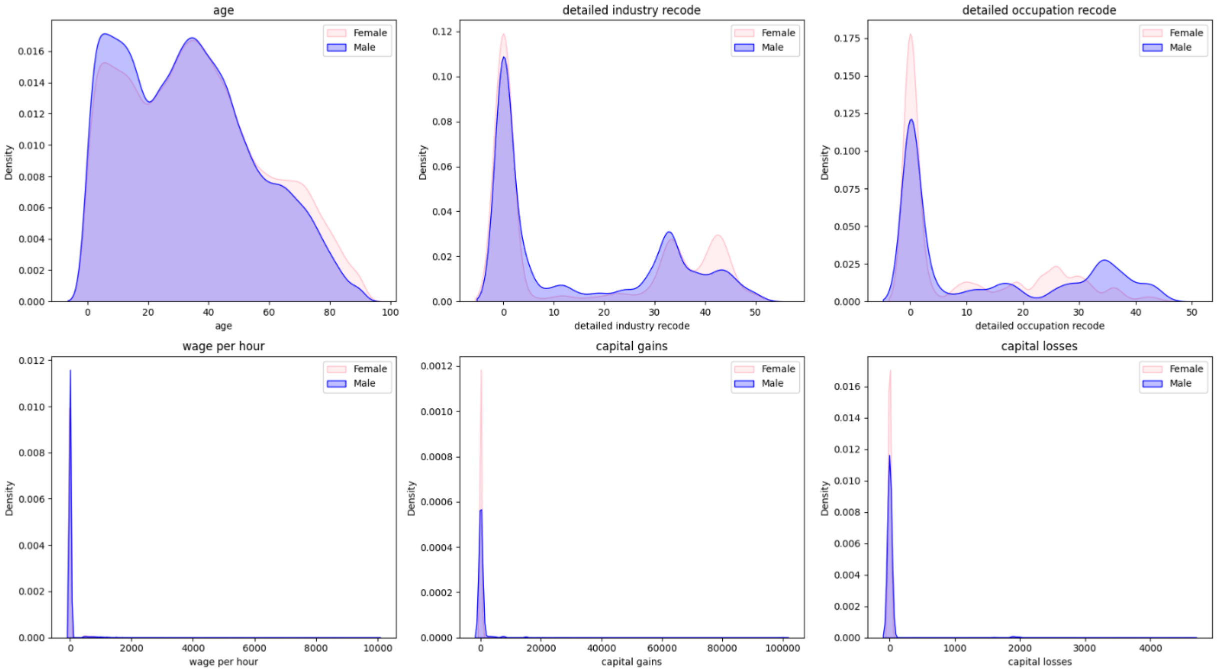
For segmentation, a K-Means clustering model was trained on preprocessed demographic and socioeconomic features. The data was grouped into seven meaningful clusters, each representing distinct socioeconomic and demographic profiles. These clusters provide insights into marketing strategy, product targeting, and customer relationship management.

2. Data Exploration and Pre-processing

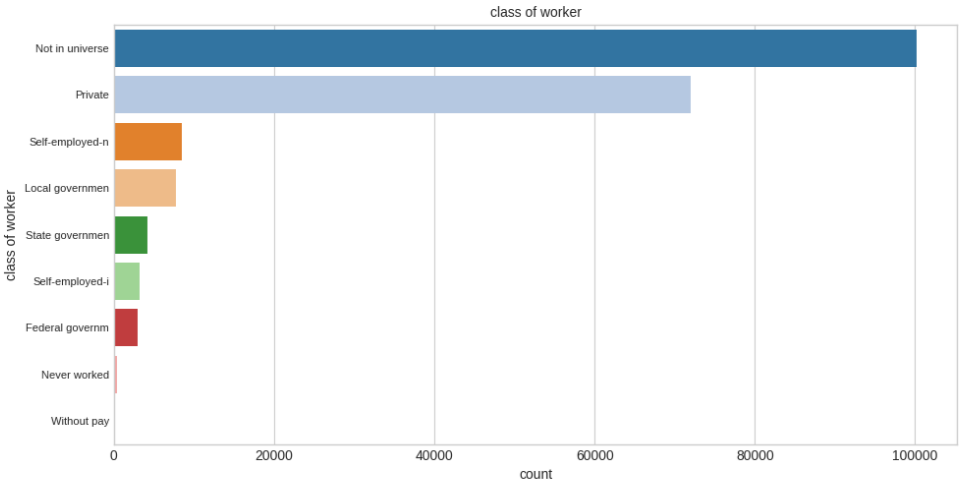
2.1. Data Overview

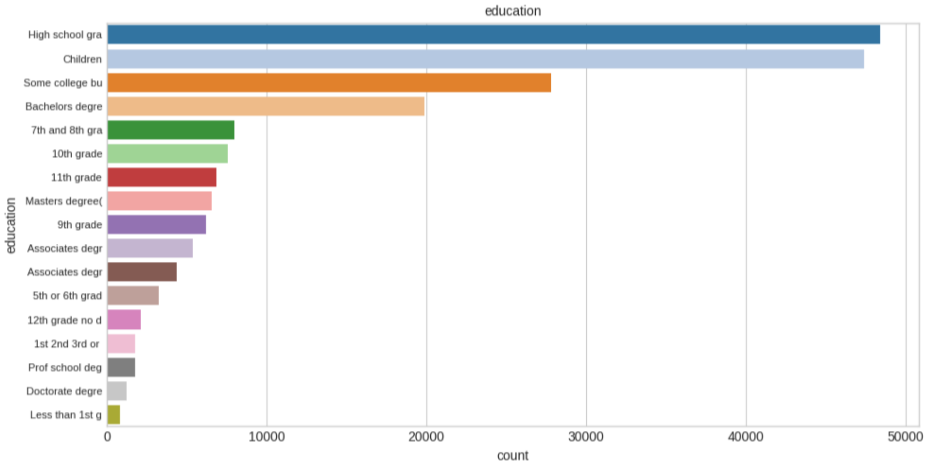
The data set contains 199523 samples from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau. Each data row contains 40 demographic and employment related variables, a weight indicating the relative distribution of people in the general population, and a label indicating whether the individual earns more than $50k or not.

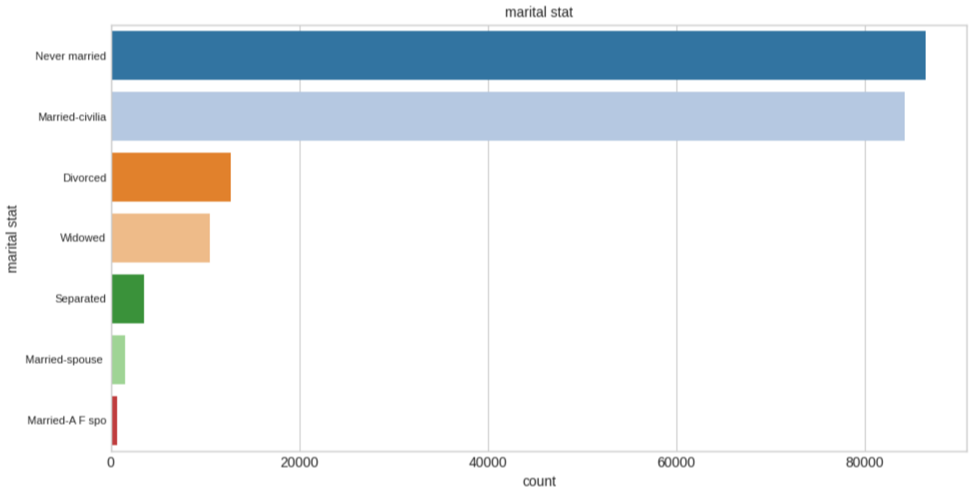
Among the 40 variables, 6 are numerical with the following distribution demonstrated by male and female categories. For the age, both male and female have a similar double-peak distribution, where most samples lie in 0-20 and 35-40. For the industry and occupation code, most samples have value 0, which indicates missing or not applicable. For wage per hour, capital gains and capital losses, most of the data are 0.



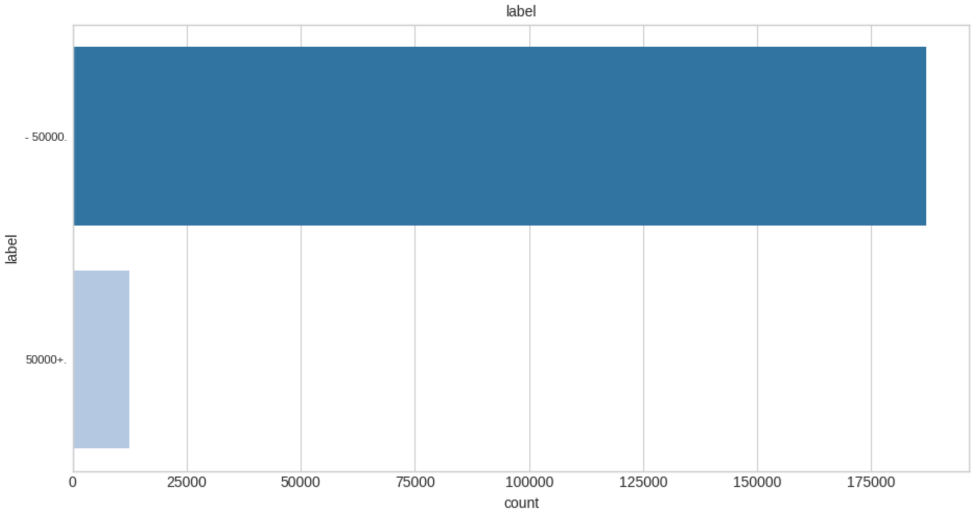
To overview, class of worker, education, marital status was shown below. A complete image of all 40 variables is available in the attachment *data distribution.png*





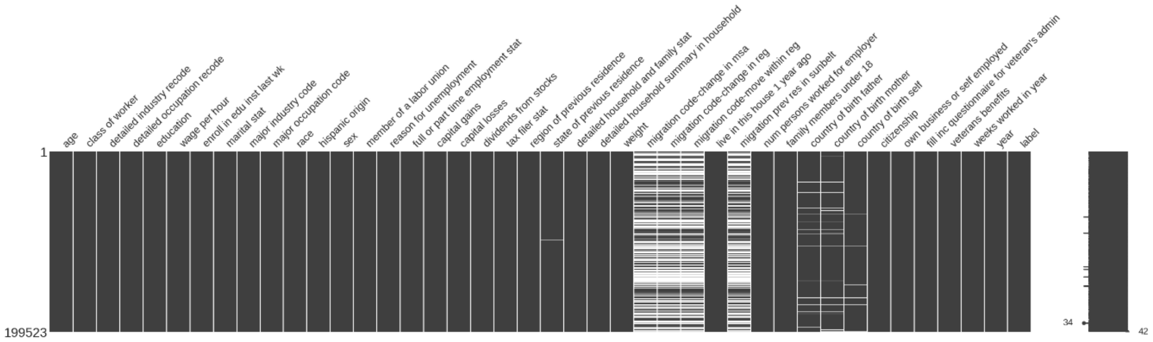


It is worth noting that only ~6% of the data are labeled as positive, which means that the data is highly unbalanced the classifier would be biased to the majority.



2.2. Pre-processing

Illustration below shows that about 50% of 4 migration related features are missing values. Therefore these columns were dropped in both the classification and segmentation to prevent noisy training data.



Numerical missing values were replaced with pd.NA, categorical missing values were replaced with 'Unknown'. Note that the 'Not in universe' values were kept as it, because they still have meaningful demographic information representing that group.

One-hot encoding was used for converting categorical variables into a vector with 1 entry of value 1, indicating its category, and other entries of value 0. This enabled interpretable predictions of high-income individuals for targeted marketing.

Note that the weight column was detached from the data and was used as the sample weight argument in the classification to reflect real-world population proportions.

For the segmentation task, numerical features were standardized to have zero mean and unit variance before K-Means clustering. This ensured that no single variable (e.g., age, wage) dominated the clustering process, allowing fair identification of customer segments.

3. Income Prediction Model

3.1. Models

In this binary classification task, 4 models were trained and evaluated.

a) Random Forest
An ensemble of decision trees, where each tree is trained on a random subset of data and features. The final prediction is based on majority voting across all trees.

b) Logistic Regression
A linear model that calculates a weighted sum of input features and applies a logistic (sigmoid) function to estimate the probability of earning.

c) XGBoost
A boosting algorithm that builds decision trees sequentially, where each new tree corrects errors from previous ones. Uses gradient descent to minimize classification loss.

d) MLP Neural Network
A feed-forward neural network with an input layer (features), one or more hidden layers (neurons with nonlinear activation functions), and an output layer (sigmoid for probability of income)

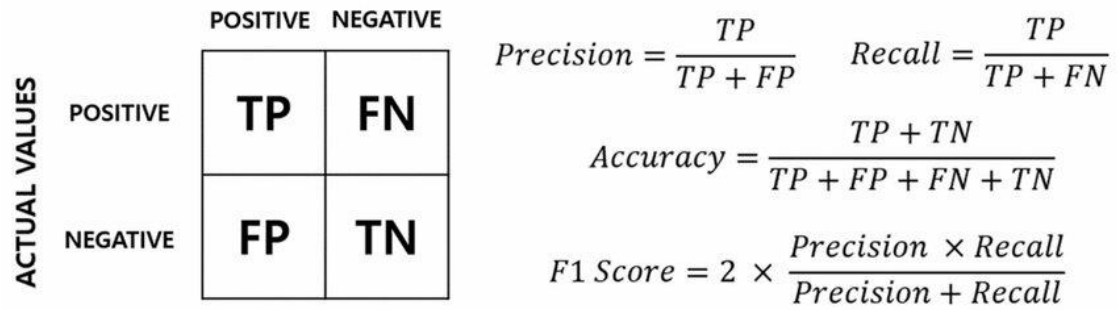
3.2. Training and Evaluation

To ensure fair and robust model assessment, the dataset was divided into training and testing subsets using an 80/20 stratified split.

During the training, sample weight was applied to random forest, logistic regression and XGBoost to reflect its distribution in the stratified sampling. To improve model performance on the unbalanced data, class weight for the label was set inversely proportional to its frequency in the training data.

For evaluation, precision, recall, F1 score, and ROC-AUC were used. Note that the accuracy was not as informational as other metrics since ~94% of the data were negative.

a) Accuracy: overall correctness of predictions.
b) Precision: the proportion of predicted high-income cases that were correct, measuring the reliability of positive predictions.
c) Recall: the proportion of actual high-income individuals correctly identified by the model, reflecting its ability to capture minority-class cases.
d) F1-score: the harmonic mean of precision and recall, balancing both false positives and false negatives.
e) ROC-AUC: measures the ability of the model to distinguish between the two income categories



		POSITIVE	NEGATIVE
ACTUAL VALUES	POSITIVE	TP	FN
	NEGATIVE	FP	TN

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1\ Score = 2 \times \frac{Precision\ \times Recall}{Precision + Recall}$$

3.3. Data Augmentation

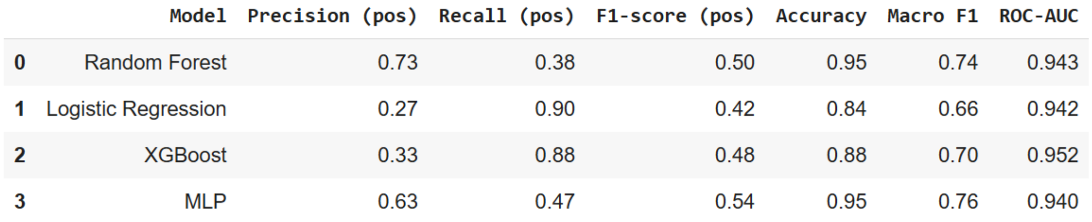
CTGAN was trained and used to generate synthetic positive data. In this step, 50000 positive samples were integrated into the training data to make the proportion of positive samples increase from ~6% to ~30%. It exposed the models with more positive data and improved the models' ability to distinguish them in the prediction task.

As mentioned in the pre-processing step, highly missing rate columns were also dropped before feeding into the CTGAN. After generating the synthetic data, the columns were restored back with default missing value to ensure consistency with the original data.

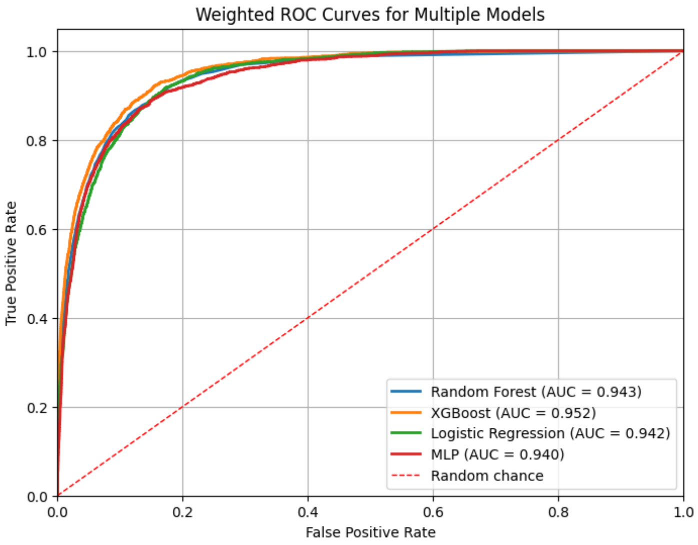
To use the synthetic data with the original data, the training data was concatenated with the synthetic data. The test data did not contain any synthetic data, leaving it genuine and interpretable.

3.4. Results

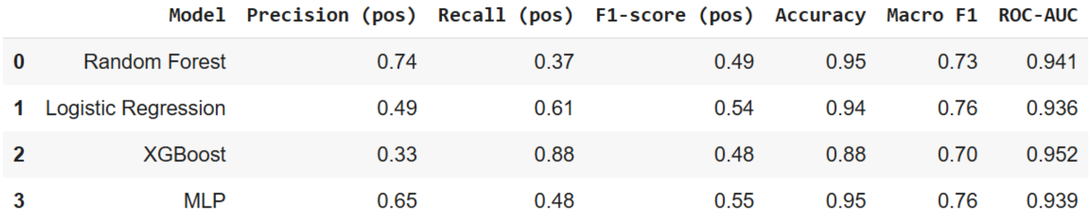
For the 4 classifiers trained on the original dataset, the evaluation results are below.



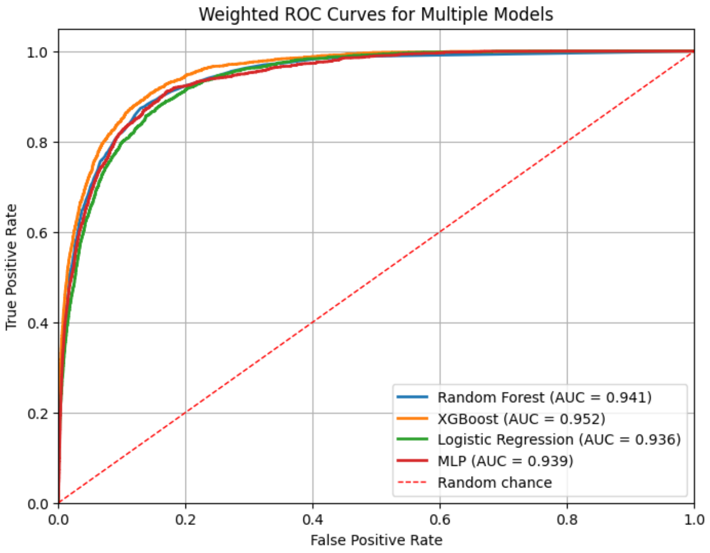
	Model	Precision (pos)	Recall (pos)	F1-score (pos)	Accuracy	Macro F1	ROC-AUC
0	Random Forest	0.73	0.38	0.50	0.95	0.74	0.943
1	Logistic Regression	0.27	0.90	0.42	0.84	0.66	0.942
2	XGBoost	0.33	0.88	0.48	0.88	0.70	0.952
3	MLP	0.63	0.47	0.54	0.95	0.76	0.940



For the 4 classifiers trained on the combined augmented dataset, the results are below,



	Model	Precision (pos)	Recall (pos)	F1-score (pos)	Accuracy	Macro F1	ROC-AUC
0	Random Forest	0.74	0.37	0.49	0.95	0.73	0.941
1	Logistic Regression	0.49	0.61	0.54	0.94	0.76	0.936
2	XGBoost	0.33	0.88	0.48	0.88	0.70	0.952
3	MLP	0.65	0.48	0.55	0.95	0.76	0.939



Both Random Forest and MLP have high overall accuracy, but low recall for the minority class. Logistic Regression and XGBoost have high recall for minority class, but have lower precision, which suggests many false positives. MLP and Random Forest have the overall best average F1 score.

For the results on the augmented data, Logistic Regression boosted recall significantly (0.61 vs 0.27 earlier), making it a strong model comparative to the MLP and Random Forest models in average F1 score. MLP also slightly benefited from the augmentation. For Random Forest and XGBoost, the results were similar, meaning that their handling of unbalanced class was good without the data augmentation.

3.5. Model Selection and Business Decision

If the business goal is broad marketing reach (casting a wide net to identify as many potential high-income individuals as possible), Logistic Regression with augmented data is recommended due to its strong recall. This would be suitable for events where the cost of false positives is relatively low.

If the business goal is precision targeting (minimizing wasted marketing spend on false positives), Random Forest or MLP are recommended, because they provide the best F1 balance and overall efficiency in identifying true high-income individuals.

In practice, we can apply hybrid strategy, where we use the Logistic Regression with augmentation to conduct an initial screening to capture a broad number of high-income individuals. Then, we can use Random Forest or MLP to refine the pool and cross-validate the predicted results for targeted recommendations.

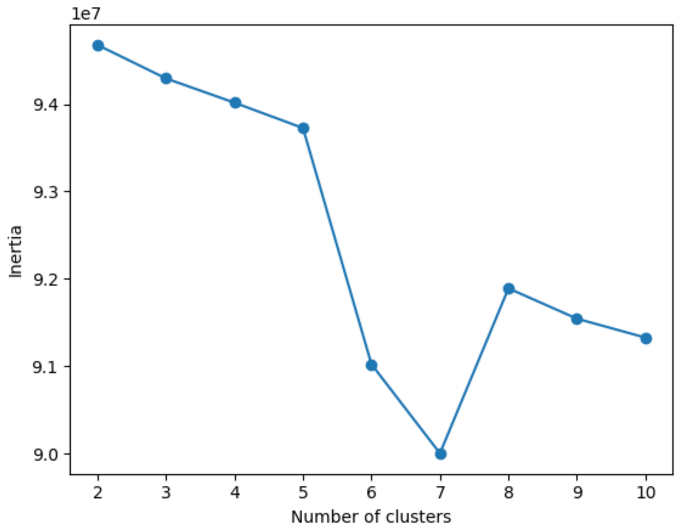
4. Customer Segmentation

4.1. Clustering model

For the customer segmentation task, a K-Means clustering model with elbow method was used.

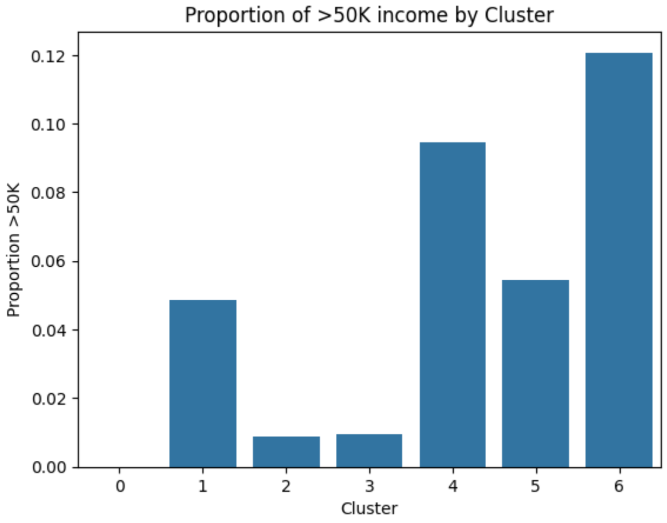
K-Means partitions the data into k clusters by minimizing the within-cluster sum of squared distances. Each point is assigned to the nearest centroid, and centroids are updated iteratively until stable. It is efficient and widely used for market segmentation.

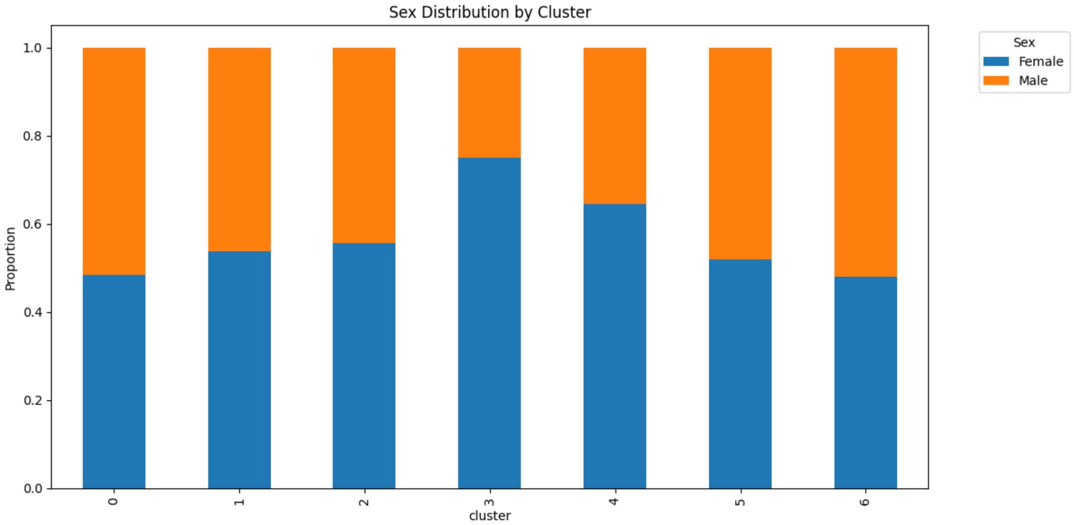
The Elbow Method helps choose k by plotting inertia (within-cluster variance) against the number of clusters. The “elbow” point is the point with the lowest inertia, indicates the clusters become smallest and tightest, which is the point for optimal clusters. In this project, the elbow occurred at k = 7, as illustrated below.

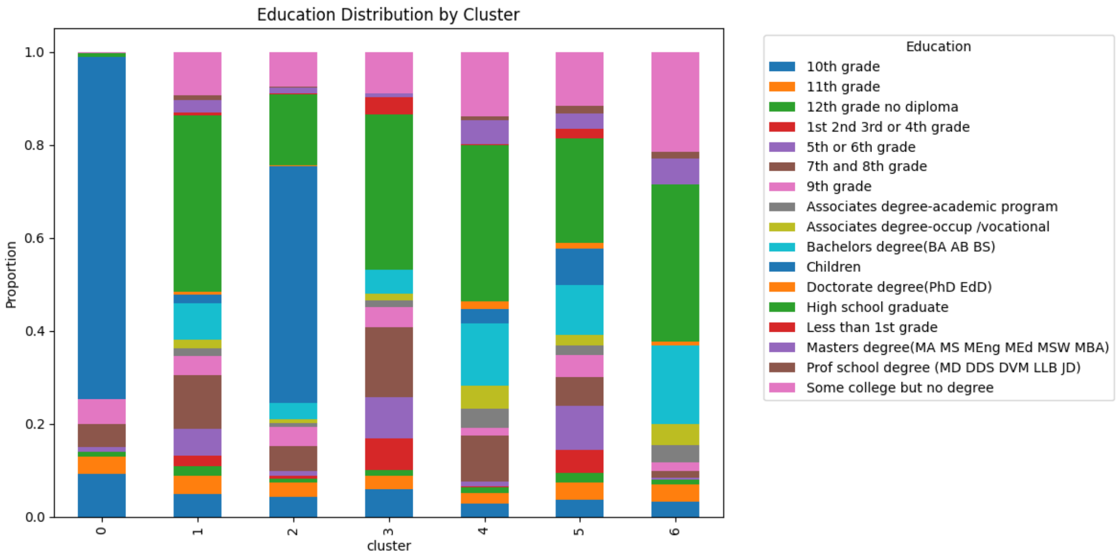


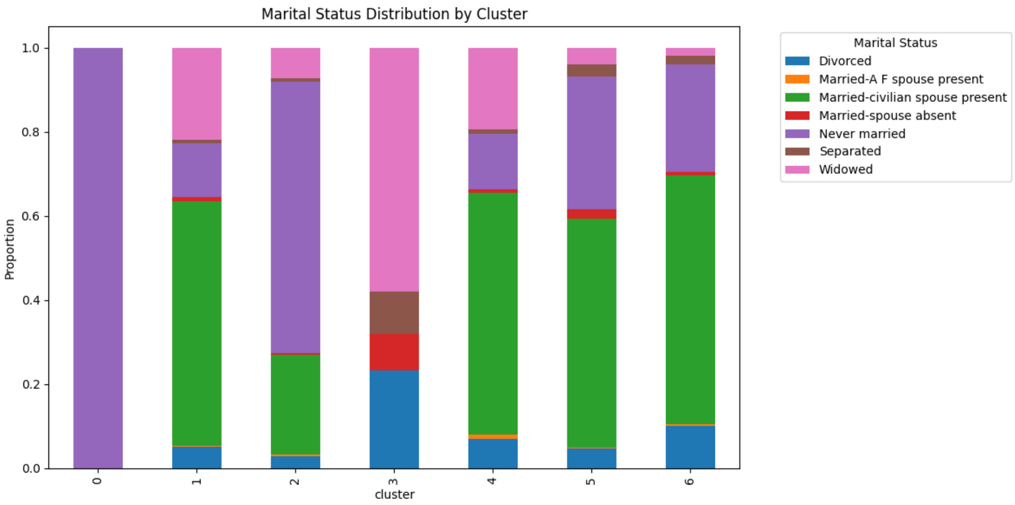
4.2. Clustering Results

The 7 groups have the following income, sex, education and marital status distributions

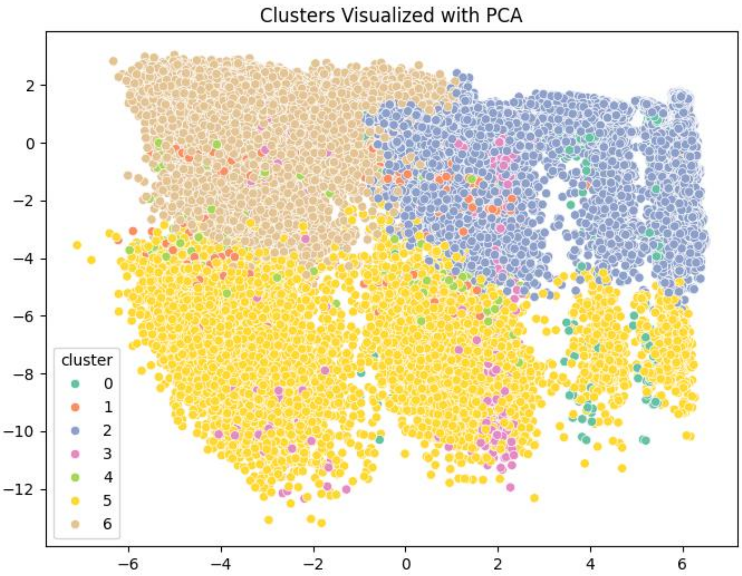




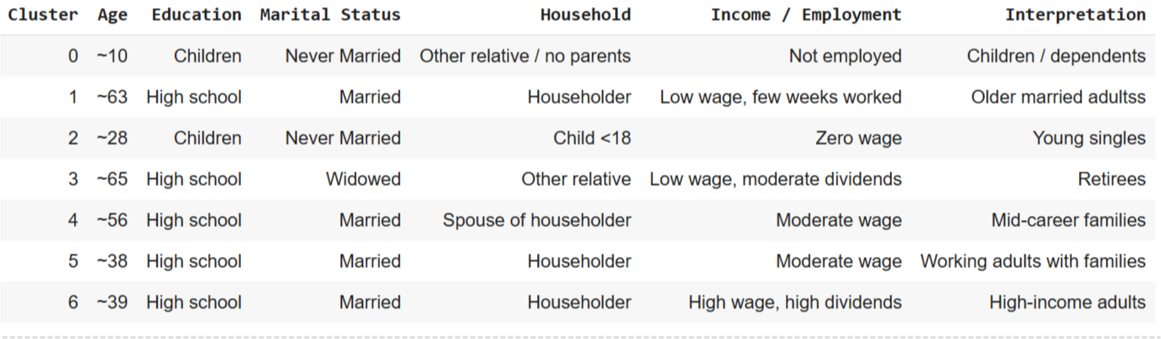




A 2D visualization with PCA of 2 leading components is illustrated as below for an overview



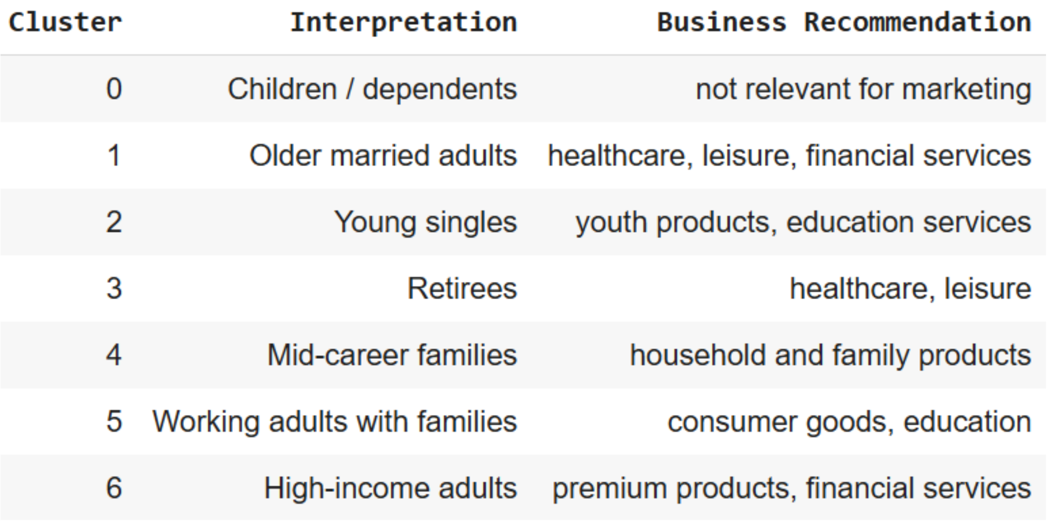
We describe the groups as the following,



Cluster	Age	Education	Marital Status	Household	Income / Employment	Interpretation
0	~10	Children	Never Married	Other relative / no parents	Not employed	Children / dependents
1	~63	High school	Married	Householder	Low wage, few weeks worked	Older married adultss
2	~28	Children	Never Married	Child <18	Zero wage	Young singles
3	~65	High school	Widowed	Other relative	Low wage, moderate dividends	Retirees
4	~56	High school	Married	Spouse of householder	Moderate wage	Mid-career families
5	~38	High school	Married	Householder	Moderate wage	Working adults with families
6	~39	High school	Married	Householder	High wage, high dividends	High-income adults

4.3. Conclusion and Business Decision

For the 7 clusters, we can target them with different products as described in the charts below



Cluster	Interpretation	Business Recommendation
0	Children / dependents	not relevant for marketing
1	Older married adults	healthcare, leisure, financial services
2	Young singles	youth products, education services
3	Retirees	healthcare, leisure
4	Mid-career families	household and family products
5	Working adults with families	consumer goods, education
6	High-income adults	premium products, financial services

This segmentation model can be used to tailor marketing campaigns by cluster, ensuring products align with the demographics, income, and household structure of each group.

In addition, we can combine segmentation with income prediction to prioritize high-value targets for selling premium products and financial services, which yields more profit.

5. Future Work

Despite using data augmentation, the class imbalance remains challenging. Further experimentation with synthetic data generation, resampling techniques, or class-weight adjustments could be explored.

Feature engineering on individual attributes may improve model performance. For instance, creating intervals for age, combining related categorical features, or transforming skewed numerical features could help. Highly correlated columns can be identified and removed to improve model robustness.

References

Xu, L., et al. *Modeling Tabular Data using Conditional GAN*, NeurIPS 2019