Project Report

Census Income Classification & Segmentation

Prepared for: J.P. Morgan Chase & Co.

ML Take-Home Assessment

Naman Deep

February 2026

1. Executive Summary

This project addresses two business tasks for a retail client using US Census Bureau Current Population Survey data (1994–1995, 199,523 records, 40 demographic and employment features):

- **Classification:** Predict whether an individual earns more or less than $50,000/year, enabling targeted marketing and customer segmentation by income level.
- **Segmentation:** Identify distinct population segments with differentiated demographic and economic profiles for tailored marketing strategies.

Key Results: For classification, XGBoost achieved a ROC-AUC of 0.949 with 87% recall on the high-income class, outperforming a Logistic Regression baseline. For segmentation, PCA dimensionality reduction followed by K-Means clustering identified 4 distinct segments ranging from children/dependents to high-earning professionals.

Note: This report emphasizes the reasoning and business judgment behind each methodological decision, not just model metrics. Where tradeoffs exist, I explain the rationale for the chosen approach.

2. Data Exploration

2.1 Dataset Overview

The dataset contains 199,523 observations with 40 features plus a sampling weight and a label indicator. Features span demographics (age, sex, race, education), employment (industry, occupation, class of worker), income components (capital gains/losses, dividends), and geographic/migration variables.

Many categorical columns are dominated by "Not in universe" — a valid Census Bureau coding indicating the question does not apply to that respondent (e.g., occupation questions for children).

Why this matters for a business client: "Not in universe" is not missing data — it is a legitimate response meaning the question is structurally inapplicable. Treating it as missing and imputing would introduce bias. Instead, I preserve it as a category, and in feature selection I drop columns where this coding dominates (>90%), since such columns carry minimal discriminative signal.

2.2 Class Imbalance

The target label is severely imbalanced:

- ≤50K: 187,141 records (93.8%)
- >50K: 12,382 records (6.2%)

This 15:1 ratio means accuracy is misleading, a trivial classifier predicting ≤50K for every record achieves 93.8% accuracy while being completely useless. Model evaluation therefore focuses on ROC-AUC (to measure ranking ability), Recall (to ensure we capture as many high-value prospects as possible), Precision (to measure the efficiency of our targeting), and the precision-recall curve, which are more informative under class imbalance.

Why not resample? I chose cost-sensitive learning (class weights) over SMOTE or undersampling. Oversampling risks overfitting to synthetic patterns; undersampling discards 93% of data. Cost-sensitive learning uses all data while adjusting the loss function, a cleaner approach for this dataset size.

2.3 Key Feature Patterns

Numeric features: capital gains, capital losses, dividends from stocks, and wage per hour are heavily zero-inflated — the vast majority of records have zero values. Binary flags (e.g., has_capital_gains) were engineered to capture the meaningful presence/absence signal in these features, while retaining the raw values for magnitude information.

Categorical features: Education, occupation, and marital status show strong differentiation between income classes. For example, individuals with professional degrees or in executive/managerial roles have substantially higher rates of >$50K income.

High correlations: *num persons worked for employer* correlates with weeks worked in year (r=0.747). Both are retained as they represent distinct concepts (employment count vs. duration). The detailed recode columns are dropped for redundancy with their major-code counterparts.

2.4 Feature Selection

Feature selection uses straightforward, interpretable EDA criteria rather than a single statistical measure. Each column is evaluated on transparent rules:

- **Single-value dominance:** if >95% of rows have the same value, the column carries almost no information
- **"Not in universe" dominance:** if >90% of rows are "Not in universe", the only signal is "applicable vs. not", already captured by other features
- **High missingness:** columns with >40% "?" values are unreliable for prediction
- **Redundancy:** a lower-cardinality alternative exists (e.g., detailed recode → major code)
- **Multiple weak signals:** combinations of moderate issues that jointly make a column unreliable

15 columns dropped → 26 original + 3 engineered = 29 features retained.

Feature importance is assessed during modeling using SHAP values (see Section 4.4) rather than during EDA, giving model-aware importance that accounts for feature interactions.

Design decision: "?" values are kept as a distinct category rather than imputed, since missingness in census data can itself be informative (e.g., refusal to report occupation may correlate with certain income patterns).

3. Preprocessing

3.1 Shared Preprocessing

- **Label encoding:** "50000+." → 1, "- 50000." → 0
- **Feature engineering:** Three binary flags (has_capital_gains, has_capital_losses, has_dividends) from zero-inflated numeric columns
- **Column drops:** 15 columns removed per data-backed EDA findings above
- **NaN handling:** 874 NaN values in hispanic origin filled with "Missing"

3.2 Classification-Specific Preprocessing

- **Encoding:** OneHotEncoder for categoricals (required for Logistic Regression's linear decision boundary; XGBoost handles it fine either way)
- **Scaling:** StandardScaler on numeric features only

- **Train/test split:** Hybrid 80/20 — all year-94 data + stratified 60% of year-95 → train (159,644 samples); remaining 40% of year-95 → test (39,879 samples)
- **Sample weights:** Census weight column used as sample_weight during model fitting to respect the stratified sampling design

Why this split strategy? A pure temporal split (train=94 (50%), test=95 (50%)) wastes 50% of data on testing. A random split ignores temporal structure. This hybrid maximizes training data while keeping the test set purely from the later year, preserving temporal validity. Stratified sampling within year-95 ensures the ~94/6 class ratio is maintained in both sets.

3.3 Segmentation-Specific Preprocessing

- **Encoding:** OrdinalEncoder for categoricals — OneHotEncoder would create ~500+ sparse dimensions that distort Euclidean distance for K-Means
- **Scaling:** StandardScaler on all features after encoding to ensure uniform scale across dimensions
- **Weight and label dropped** from features; label is used for post-hoc cluster profiling only

4. Classification Model

4.1 Model Selection Rationale

Logistic Regression (baseline): An interpretable linear model. Serves as a baseline to quantify the marginal value of a more complex model.

XGBoost (primary): Gradient-boosted decision trees — the established standard for tabular classification tasks. Handles feature interactions naturally and is robust to outliers. Configured with 300 estimators, max depth 6, learning rate 0.1.

Why these two? Starting with a linear baseline and a tree-based model covers the two most common failure modes: a problem that's inherently linear (where LR suffices) vs. one with complex interactions (where XGBoost excels). The gap between them reveals how much interaction modeling matters for this dataset.

Both models are trained with census sample weights to produce population-representative predictions.

4.2 Handling Class Imbalance

Rather than resampling (which discards data or creates synthetic examples), imbalance is handled through cost-sensitive learning:

- **Logistic Regression:** class_weight="balanced" — sklearn automatically adjusts weights inversely proportional to class frequencies
- **XGBoost:** scale_pos_weight = (1 − 0.061) / 0.061 ≈ 15.4 — explicitly scales the positive class gradient contribution

4.3 Results

Metric	Logistic Regression	XGBoost
ROC-AUC	0.9448	0.9493
Precision (>50K)	0.29	0.34
Recall (>50K)	0.90	0.87
F1 (>50K)	0.44	0.48

Key observations:

- Both models achieve ~0.95 ROC-AUC, indicating strong ranking ability — the model reliably assigns higher probability scores to actual high-income individuals.
- Logistic Regression achieves very high recall (0.90) at the cost of lower precision (0.29); it catches most high-income individuals but with many false positives.
- XGBoost provides a better precision-recall tradeoff: 87% recall with 34% precision. This means fewer wasted marketing dollars on misclassified individuals.
- The test set is purely year-95 data, preserving temporal validity; the model is never tested on same-year data it trained on from year-94.

Honest assessment: The precision (>50k) of 0.34 means roughly 2 in 3 flagged individuals are actually ≤50K. This is a direct consequence of the 15:1 class imbalance. For high-cost campaigns, the probability threshold should be raised to improve precision at the cost of some recall. This ensures marketing dollars are spent only on the most likely high-value customers, reducing waste and improving ROI, an essential discipline when each outreach carries a meaningful cost.

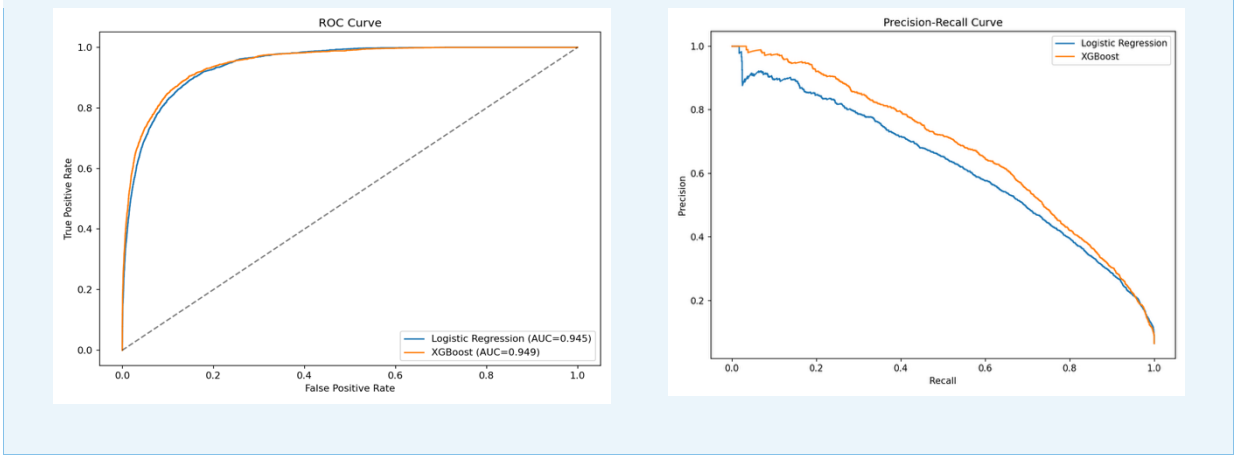

Figure: ROC & Precision-Recall Curve — Logistic Regression vs. XGBoost

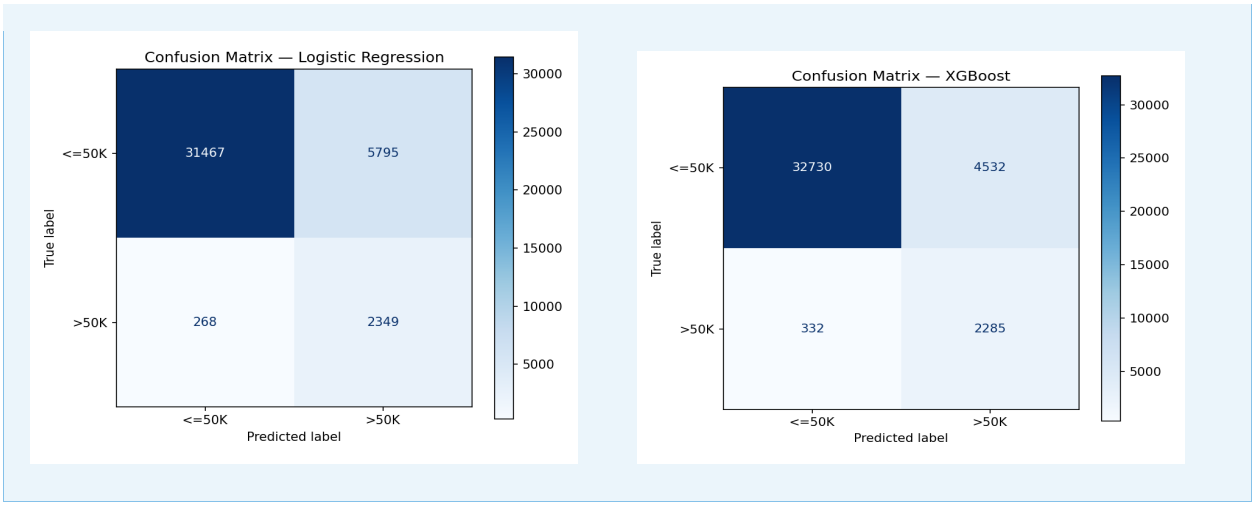

Figure: Confusion Matrix — Logistic Regression vs XGBoost

4.4 Feature Importance (SHAP Analysis)

Feature importance is assessed using SHAP (SHapley Additive exPlanations) values, which provide model-aware importance that accounts for feature interactions — unlike simple gain-based importance which can be misleading for correlated features.

Top 10 Features by Mean |SHAP Value| (XGBoost):

Rank	Feature	Mean |SHAP|
1	age	1.818
2	weeks worked in year	1.013
3	tax filer stat = Nonfiler	0.709
4	sex = Female	0.441
5	dividends from stocks	0.287
6	num persons worked for employer	0.223
7	capital gains	0.172
8	education = Bachelors degree	0.164
9	detailed household summary = Householder	0.158
10	education = High school graduate	0.120

Interpretation: The top features align with economic intuition — income correlates strongly with work experience (age, weeks worked), education level, and investment activity (dividends, capital gains). The tax filer status serves as a strong proxy for overall financial engagement. The SHAP beeswarm plot below shows both the magnitude and direction of each feature's impact on individual predictions.

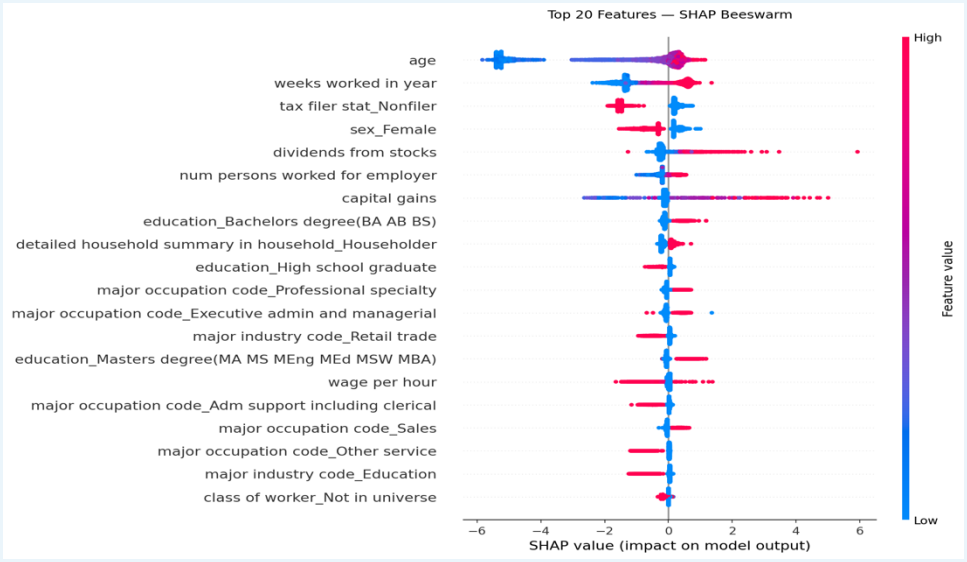

Figure: SHAP Beeswarm — Feature impact direction and magnitude

5. Segmentation Model

5.1 Approach

Dimensionality Reduction: PCA reduces the 29 preprocessed features to 17 components capturing 85% of variance. This is critical for K-Means, which suffers from the curse of dimensionality — Euclidean distance metrics become less meaningful in high-dimensional space, causing all points to appear equidistant.

Clustering: K-Means with silhouette score evaluation across k=2 to k=7. The optimal k=4 was selected based on the highest silhouette score (0.23). While this score is typical for real-world demographic data, it indicates that the resulting segments are distinct enough to support differentiated marketing strategies, without being overly fragmented.

Why K-Means over alternatives? For a business client, K-Means produces clean, non-overlapping segments that are easy to operationalize. DBSCAN would require density tuning and produces irregular-shaped clusters harder to action. Gaussian Mixture Models provide probabilistic assignment but add complexity without clear business benefit here.

5.2 Silhouette Analysis

Silhouette analysis provides per-cluster quality assessment beyond the aggregate score. For each value of k, blade plots show the distribution of silhouette coefficients within each cluster, clusters with many negative-silhouette points are poorly defined. From a business perspective, higher silhouettes indicate clean, reliable segments that support precise targeting, while lower or negative silhouettes signal areas where segments overlap and marketing or product decisions should remain broader or less personalized.

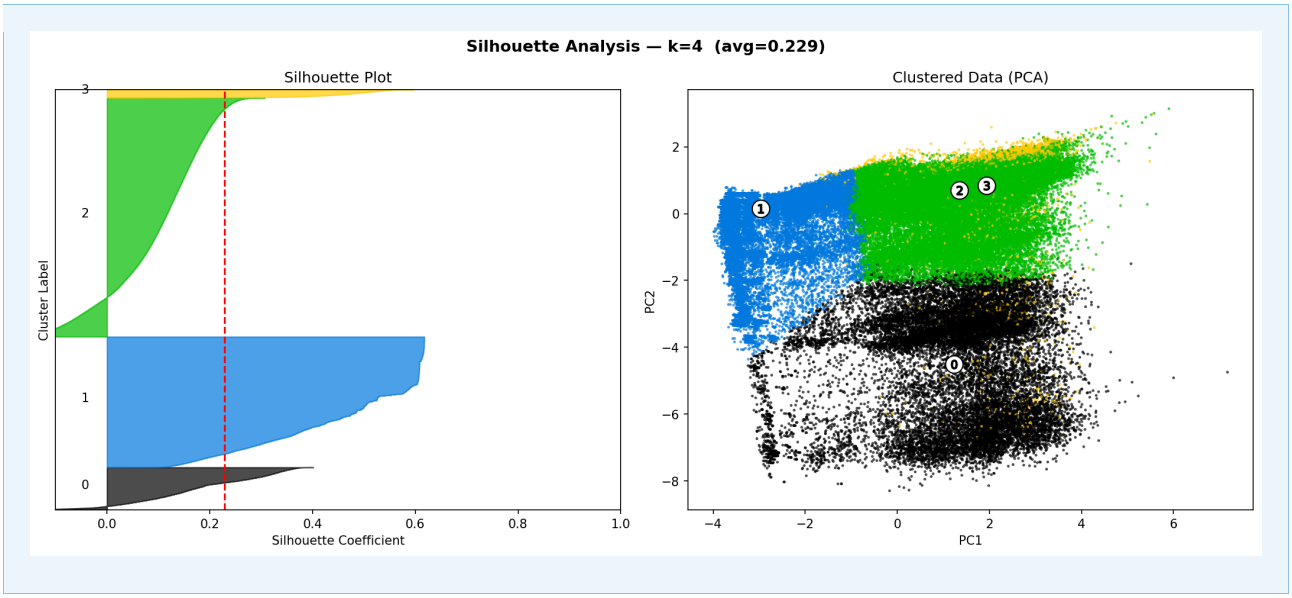

Figure: Silhouette Analysis — k=4 (selected)

5.3 Segment Profiles

Cluster	Size	Mean Age	>50K Rate	Key Characteristics
0	31.1%	10.3	0.0%	Children/youth, not in workforce, never married
1	2.0%	44.2	29.9%	High-earning males, married, private sector workers
2	56.8%	46.0	8.9%	Core working adults, married, mixed occupation, female-leaning
3	10.1%	42.5	5.4%	Lower-income working adults, mixed employment

5.4 Segment Interpretation for Marketing

The 4 clusters form clean, actionable marketing tiers:

1. Premium Segment (Cluster 1): 2% of population, ~30% earn >$50K. Predominantly male, married, private sector. Target with premium financial products, investment services, wealth management offerings.

2. Core Working Adults (Cluster 2): 57% of population, 8.9% earn >$50K. Largest segment — married adults across industries. Target with everyday retail, savings products, insurance, and career development services.

3. Budget-Conscious Workers (Cluster 3): 10% of population, 5.4% earn >$50K. Lower income, mixed employment patterns. Target with value-oriented products, education/upskilling programs, and budget financial tools.

4. Youth/Dependents (Cluster 0): 31% of population, near-zero income. Children and young dependents. Target parents of this segment with family-oriented products and education savings plans.

Categorical Profile Distributions

The following charts show how key demographic attributes distribute across the 4 segments:

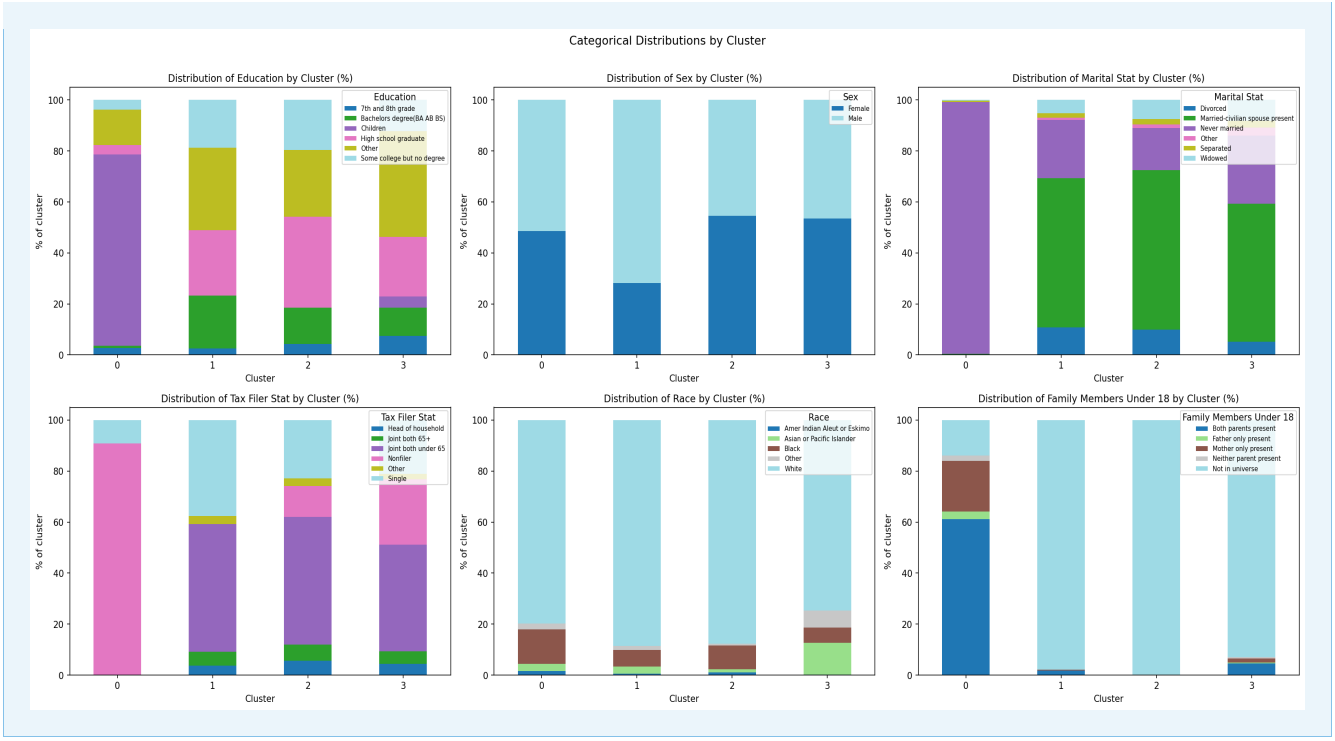

Figure: Categorical Distribution by Cluster

6. Business Recommendations

6.1 Classification Deployment

- Deploy XGBoost as the primary income classifier for targeting high-value customers.
- At the default threshold, the model identifies 87% of high-income individuals with 34% precision — suitable for broad marketing campaigns where reaching most high-income individuals matters more than avoiding false positives.
- For targeted, high-cost campaigns (e.g., premium product launches), raise the probability threshold to improve precision at the cost of some recall. A cost-benefit analysis specific to campaign economics should guide this threshold.
- Consider A/B testing model-targeted vs. untargeted campaigns to quantify lift in conversion rates.

6.2 Segmentation Deployment

- Use segments for differentiated messaging: premium financial products for Cluster 1, everyday retail for Cluster 2, value-oriented products for Cluster 3, family/education products for Cluster 0.
- Combine classification and segmentation: within each segment, use the classifier's probability scores to further prioritize outreach. This creates a two-dimensional targeting framework (segment × income likelihood).
- Monitor segment stability over time — demographic shifts may require periodic re-clustering.

6.3 Deployment and Monitoring Guidance

- Retrain models quarterly or when new census data becomes available.
- Monitor for data drift — feature distributions may shift year over year.
- Include a human review step for high-stakes decisions (credit, lending) to avoid bias amplification.
- Implement logging of model predictions and outcomes to enable ongoing performance tracking and threshold calibration.

7. Open Questions and Considerations

Several questions would benefit from client input before production deployment:

- **Cost matrix:** What is the relative cost of a false positive (marketing to a low-income individual) vs. a false negative (missing a high-income individual)? This directly determines the optimal classification threshold.
- **Campaign granularity:** Are the 4 segments sufficient, or would finer segmentation (k=5 or k=6) provide more actionable differentiation? The silhouette scores for k=5 and k=6 are close to k=4.
- **Fairness requirements:** Are there regulatory or internal fairness constraints on how sex and race features should be used in income prediction? The SHAP analysis shows sex=Female as the 4th most important feature.
- **Temporal scope:** The data is from 1994-1995. If deployed on current census data, feature distributions (especially income thresholds and occupation categories) will have shifted. A recalibration study would be advisable.

8. Future Work

- **Additional models:** LightGBM, CatBoost, or neural networks for potential performance gains. CatBoost in particular handles categorical features natively.
- **Hyperparameter tuning:** Bayesian optimization or grid search over XGBoost hyperparameters (currently set to reasonable defaults, not tuned).
- **Feature engineering:** Interaction features (e.g., education × occupation), age bins, geographic aggregations.
- **Threshold optimization:** Tune classification threshold based on business cost matrix (cost of false positive vs. false negative).
- **Fairness analysis:** Evaluate model performance across protected groups (sex, race) to ensure equitable predictions and compliance with fair lending regulations.
- **A/B testing:** Validate segment-based marketing strategies through controlled experiments before full rollout.
- **Cluster stability:** Use bootstrap resampling to assess how stable clusters are across different data samples.
- **Real-time scoring:** Package the XGBoost model as a REST API endpoint for integration with CRM and marketing automation systems.

9. References

- US Census Bureau. Current Population Survey Technical Documentation. https://www.census.gov/programs-surveys/cps/technical-documentation.html
- XGBoost documentation: https://xgboost.readthedocs.io/
- scikit-learn documentation: https://scikit-learn.org/stable/
- scikit-learn. Selecting the number of clusters with silhouette analysis on KMeans clustering. https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html