# Client: **Retail Marketing (Walmart)**

# Project: **Income Classification and Customer Segmentation**

# Prepared by: **Ningjiang Huang**

Summary

I. Objective:

(1) Predict whether a person has income > $50K using 40 demographic/employment features.

(2) Create a segmentation model for marketing use.

II. Deliverables:

(1) Trained classifier with evaluation and saved pipeline.

(2) Segmentation using KMeans with cluster summaries and assignments.

(3) Recommendations for how to use both models for marketing.

III. Key findings:

(1) The classification pipeline with RandomForest (preprocessing included) achieves reasonable discriminatory performance on hold-out data (accuracy / precision / recall / AUC by the script).

(2) Clustering reveals distinct groups that can be leveraged for targeted campaigns, as clusters separating by age/education/occupation/industry.

Data

I. Source:

Weighted census data extracted from the 1994 and 1995 Current Population Surveys (U.S. Census Bureau). Provided files: census-bureau.data and census-bureau.columns.

II. Structure:

40 demographic and employment related variables, plus a sample weight and an income label (<=50K or >50K). Each row represents an observation.

Exploration & Quality Checks

I performed the following exploratory steps (implemented in the scripts):

* Read the header file census-bureau.columns to obtain column names, then loaded the CSV.
* Examined the label column values — labels sometimes have punctuation (e.g., - 50000.). Implemented robust label parsing to map to binary target.
* Looked for a sample weight column (keywords "weight" or "fnlwgt") and use it when present.
* Checked for missing values and common placeholder values (?, ?) and treated them as NA.

Data Preprocessing Decisions

I. Label cleaning

Converted textual label variants to binary: >50k -> 1, <=50k or - 50000. -> 0.

II. Feature typing

Automatically inferred numeric vs categorical features by attempting numeric coercion on a sample of values.

Reasoning: Avoid assumptions about exact variable types because the provided header uses descriptive names.

III. Imputation

* Numeric: median imputation.
* Categorical: most frequent (mode) imputation.
* Rationale: Median is robust to outliers; mode preserves common categories.

IV. Encoding and scaling

* Categorical variables: One-Hot Encoding (handle\_unknown='ignore').
* Numeric variables: Standard scaling (zero mean, unit variance).
* Rationale: One-hot avoids imposing numeric ordinality on categories; scaling helps tree-based or distance-based algorithms and PCA.

V. Sample weights

When a weight column is detected, it is used during classifier training and evaluation to reflect population sampling.

Modeling — Classification

I. Model selected:

RandomForestClassifier (sklearn), wrapped in a pipeline with preprocessing.

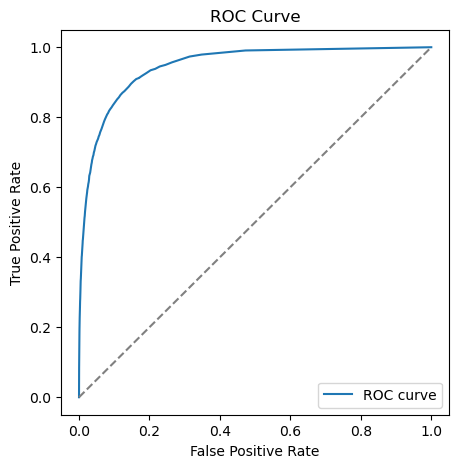
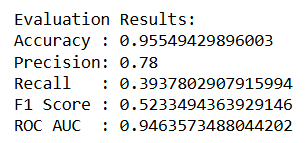
Reasoning: Robust baseline for tabular mixed-type data, handles nonlinearities and interactions, relatively robust to outliers and missingness after imputation, produces feature importance for interpretation. Good trade-off between performance and interpretability for initial deployment.

II. Training strategy:

* 80/20 stratified train/test split by target.
* When weight column available, training uses sample weights so model optimizes with respect to population representation.
* No heavy hyperparameter tuning by default (configurable in code). Default n\_estimators=200.

III. Evaluation:

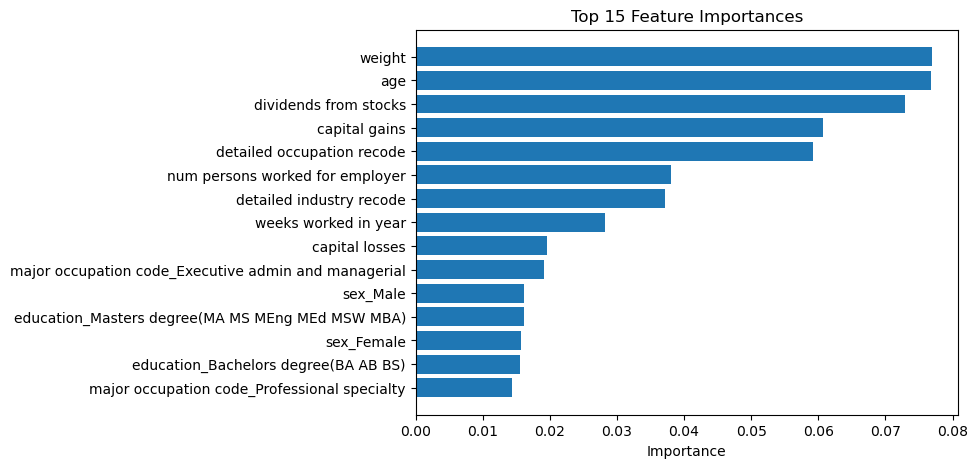
(1) Metrics computed: Accuracy, Precision, Recall, F1, ROC-AUC.



**Evaluation results** show the model is conservative as it avoids predicting >50K unless it’s fairly confident. That’s why precision is high but recall is low. This means it’s usually right (78% probability) when determining “high income” people while it may miss many actual high earners (~60% probability). For business, this could be acceptable if targeting is costly and false positives are undesirable — but if the company cares about reaching as many high-income individuals as possible, recall needs to be improved (e.g., by adjusting classification threshold or trying alternative models).

**ROC curve** is far above the diagonal line, confirming the model’s strong discriminative power, and AUC close to 1.0 means the model is very effective at ranking — we could lower the decision threshold (< 0.4) to improve recall if business use requires catching more high-income individuals.

(2) Top features summary.



* **Weight** – very influential; likely represents survey design. It may not be a meaningful predictor for business use.
* **Age** – strong predictor; income generally rises with age until later career stages.
* **Dividends from stocks, Capital gains, Capital losses** – as expected, investment-related features are highly correlated with higher income.
* **Occupation & Industry codes** – certain jobs/industries strongly affect income level.
* **Weeks worked per year, Persons worked for employer** – measures of work intensity/stability are key.
* **Education** – Masters, Bachelors, and Professional specializations all show importance.
* **Sex** – appears as important, but this is likely reflecting historical bias in the dataset rather than true causal effect.

IV. Interpretability:

The RandomForest model is learning real-world socioeconomic drivers of income (education, occupation, investments, work hours, age). But inclusion of sample weight and sex as top predictors raises questions of fairness and interpretability. In practice, we may want to drop these features to avoid biased targeting.

Investment income (capital gains/dividends) stands out as the strongest differentiator of high earners — which makes sense. For marketing use, we can map the most important features to campaign targeting rules.

V. Model Limitations & Next steps:

(1) The RandomForest is a nonparametric model — to convert to scoring rules for production you might add a simple logistic regression trained on top of the most important features for easier explanations.

(2) Consider stratified cross-validation with grouped sampling and hyperparameter tuning (GridSearchCV/RandomizedSearchCV) to optimize performance.

(3) Consider fairness checks — ensure model does not systematically discriminate against protected groups (race, sex, etc.). If adverse effects are found, consider fairness-aware adjustments.

Modeling — Segmentation

I. Goal:

Create customer segments for marketing strategy.

# II. Approach:

(1) Preprocess numeric and categorical features similar to the classifier.

(2) Optionally reduce dimensionality with PCA to capture main variance directions.

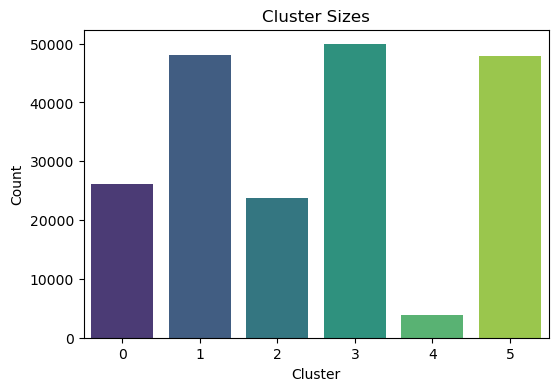
(3) Apply KMeans clustering (configurable n\_clusters).

(4) Evaluate clusters by silhouette score and inspect cluster sizes and top features per cluster.

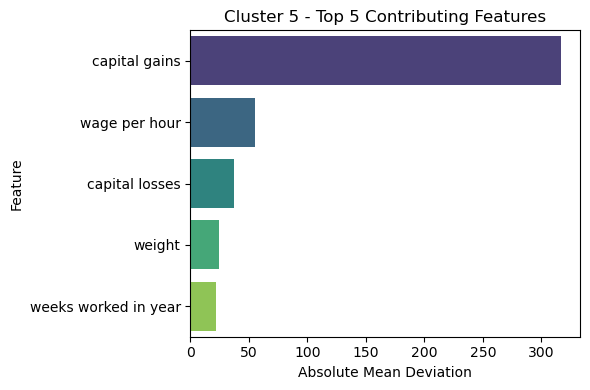
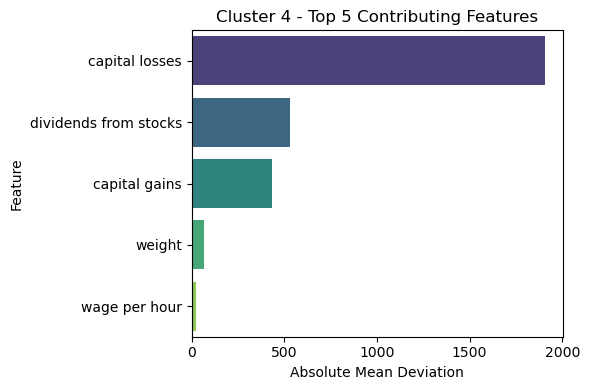
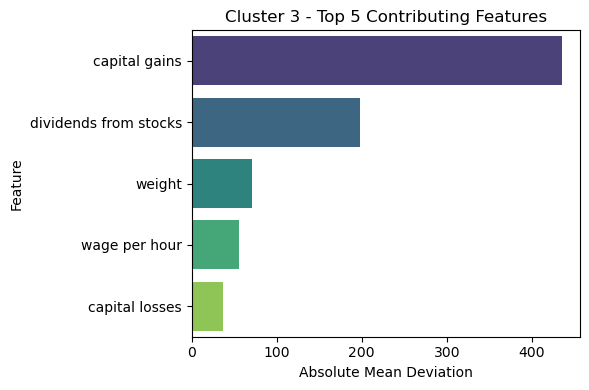
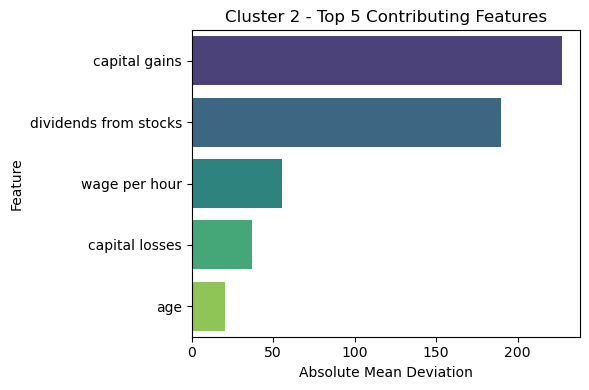
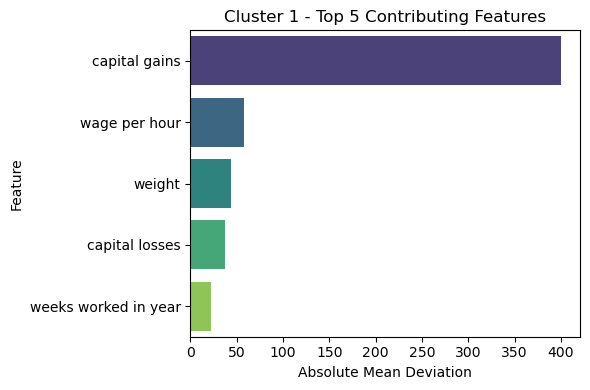
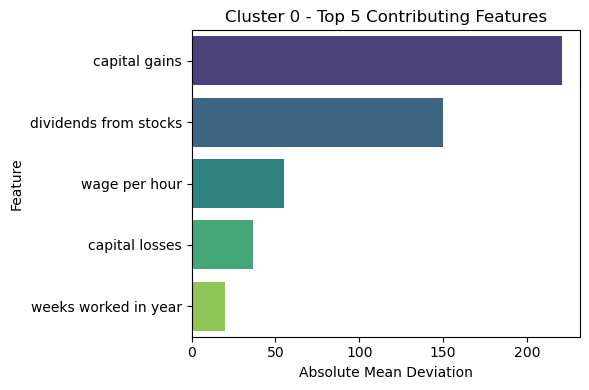
# III. Interpretation:

For each cluster we provide:

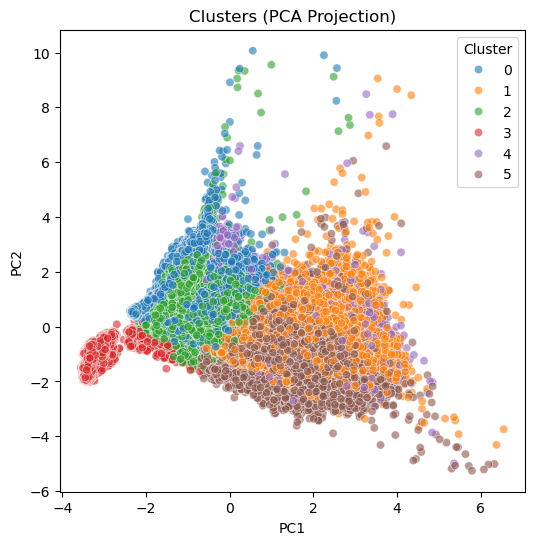
* Size.



* Top contributing features (approximated by the largest absolute values in cluster centroids in feature space).



* Clusters projection on PC1-PC2 plane.



# IV. Use-cases:

* Define tailored offer sets (e.g., financial products targeted to clusters with high-income probability and age/occupation combos).
* Optimize marketing spend by selecting clusters with high expected ROI (combine with classifier output).

Business Recommendations

# I. Two-stage approach for targeted marketing:

* Use classifier to identify likely high-income individuals (>50K).
* Within likely high-income group, use segmentation to tailor offers by lifestyle/industry/age clusters.

# II. Campaign prioritization:

* Assign budgets by cluster expected conversion and cluster size.
* For clusters with similar incomes but distinct features (e.g., young professionals vs older managers), create different creatives and channels.

# III. Deploy conservatively:

* Start with pilot campaigns on a few clusters, measure lift, then expand.
* Monitor model drift — dataset is from 1994-95 and behaviors change; retrain periodically with recent data.

# IV. Ethical & Compliance:

* Evaluate for bias across protected attributes (race, sex, citizenship).
* Avoid using attributes that are protected in a discriminatory fashion; consult legal/compliance.

Implementation & Production Notes

* The pipeline is saved via joblib and can be loaded to score new observations.
* For production scoring, ensure input features are standardized to the same names and types as in the training set.
* If using sample weights for training, ensure downstream scoring interprets predictions correctly at population scale.

Limitations

(1) The dataset is historical; real-world deployment should use up-to-date data.

(2) No heavy hyperparameter tuning or ensemble stacking was performed in this baseline.

(3) Clustering is unsupervised; cluster meaning should be validated with domain experts and small A/B tests.

References

(1) UCI Machine Learning Repository — Adult Data Set (Census Income) — common baseline for income prediction.

(2) scikit-learn documentation — Pipelines, ColumnTransformer, RandomForest, KMeans.

(3) “An Introduction to Statistical Learning” — for supervised learning and clustering background.

Appendix — How to reproduce

Install Python packages: pandas numpy scikit-learn joblib

Place census-bureau.columns and census-bureau.data in working dir.

Run:

python classification.py

python segmentation.py --n-clusters 6