Adult Census Income Dataset Analysis

Matthew Lew

Jessica Wang

April 5, 2025

Table of Contents

Dataset Description and Objective:

We decided to pick the following dataset: https://archive.ics.uci.edu/dataset/2/adult

The dataset is titled the ‘adult’ dataset, taken from a 1994 U.S. Census, it contains 48,842 instances as well as 14 features. The problem for this dataset is to use the 14 features to predict whether an individual’s income is above or below 50,000 USD a year.

Dataset Details:

Number of instances: 48,842

Number of attributes: 14

Target Variable: income (>50k, <=50k)

Attributes:

Variable Name	Variable Type	Description/Possible Values
age	Integer	An adult’s integer age
workclass	Categorical	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt	Integer	Final weight, the number of people in the U.S. population that the record represents
education	Categorical	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num	Integer	Numeric representation of the education categorical variable - the number of years of education associated with a person’s schooling
marital-status	Categorical	Married-civ-spouse, Divorced,

		Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation	Categorical	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship	Categorical	Wife, Own-child, Husband, doNot-in-family, Other-relative, Unmarried.
race	Categorical	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex	Binary	Female, Male.
capital-gain	Integer	Money gained from capital investments and assets
capital-loss	Integer	Money lost from sale of assets
hours-per-week	Integer	Number of hours worked per week at the person’s main job.
native-country	Categorical	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Task 1 - Explorative Analysis (EDA) and Data Preprocessing

Before applying any analysis or feature selection algorithms, we can make some initial observations and explorative analysis to refine our data and ensure it is ready for prediction and selection algorithms. Below are the observations we encountered during preprocessing and the steps we took to refine our data.

1. Dropping Unneeded Features:

Based on the description, as well as the data in the dataset, we can decide to prune the following features from our analysis:

fnlwgt - The number of people in the U.S. population that a record represents

We will decide to drop this column as it represents census sampling weight, and will not help with prediction of income as (continue)

education - The level of education that a person has comleted

We will decide to drop this column as it is a categorical variable that represents the level of education a person has completed, however we also have the education-num feature which is an integer value that represents the same data, so we will only keep education-num to make the data more clean.

2. Dropping Rows With Missing Or Unknown Values:

Within the data there are a large number of placeholder '?' values when there is either unknown or missing data, in total there were:

- 1836 missing workclass values
- 1843 missing occupation values
- 583 missing native-country values
- 2399 total rows being affected

Our options were to either continue analysis with missing values, impute the values with the most frequent value for that category, or remove the row altogether. Because the dataset has a total of 32561 records, simply removing the 2399 affected rows would still leave us with a large amount of data to analyze, as a result we delete these rows, and proceed with a dataset of 30162 records.

3. Class Imbalance Problem:

Out of the 30162 records, 22654 had an outcome of <=50k, and 7508 had an outcome of >50k, meaning there was an approximate 75% to 25% class imbalance, meaning that theoretically a model could predict <=50k for every instance and achieve a 75% accuracy, as a result, we decided to do further preprocessing to handle this issue.

We addressed this using two separate fixes, first we used class weights in our models, which allows the classifier to penalize misclassification of the minority class more heavily.

We also implemented SMOTE oversampling (Synthetic Minority Over-sampling Technique) to see if synthetic data points would help improve minority class recall.

After implementing our 5 different algorithms, we decided to test them with and without the class weights and SMOTE and found the following results:

Model Performance Without Class Balancing (Measured in %)				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.822476	0.741834	0.460131	0.567971
Random Forest	0.848500	0.726804	0.645098	0.683518
Naive Bayes	0.796784	0.712885	0.332680	0.453654
Support Vector Machine	0.849163	0.783364	0.560131	0.653201

Model Performance Without Class Balancing (Measured in %)				
k-Nearest Neighbour	0.832422	0.688453	0.619608	0.652219

Model Performance With Class Balancing (Measured in %)				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.771921	0.534968	0.769935	0.631297
Random Forest	0.837726	0.676038	0.691503	0.683683
Naive Bayes	0.815846	0.706811	0.467974	0.563114
Support Vector Machine	0.787005	0.551277	0.860784	0.672110
k-Nearest Neighbour	0.798110	0.574713	0.784314	0.663350

After applying class balancing techniques we observed a notable improvement in recall across all the models, most notably 46% to 77% in logistic regression, and 56% to 86% in SVM. This means that our models became much significantly better at predicting when an individual was earning our minority class of >50k. Additionally our F1 score improved for all models, showing a better balance of precision and recall. However, this came with a slight reduction in accuracy and precision for 4 of the 5 models, but this can be expected due to the increased sensitivity of the minority class, and was a negligible difference and acceptable trade-off with the increase in recall and F1 score. Overall, the increase in recall and F1 score suggest that our application of the class balancing methods made the model more effective overall, even at a minor cost of accuracy and precision.

Task 2 - Algorithm Comparison Without Feature Selection

In order to address the classification problem of prediction whether an individual's annual income is likely to exceed $50k, we tested and compared 5 different algorithms, without feature selection, each from a different category, in order to see which one was the most accurate or best overall. The algorithms that we chose were: logistic regression (linear model), random forest (bagging method). Naive Bayes (probability method), support vector machine (SVM, margin-based classifier), and k-nearest neighbours (k-NN, distance based method).

Before the data was used to train and test the models, we applied several preprocessing steps such as removing rows with missing or unknown values, and dropping features that were deemed unnecessary for our analysis. This left us with about 30000 rows left in the dataset.

After the data was trimmed, we encoded the categorical variables using LabelEncoder in order to ensure that all features were numerical (0, 1, for true, false, etc) and the features were standardized using StandardScaler in order to normalize the input space (important for certain algorithms used such as SVM and k-NN).

Class balancing techniques were applied to ensure less misclassification of the minority class. The target variable (income) was binary, with 0 representing an annual income that was below $50k and 1 representing an annual income that above $50k.

The dataset was then split into training and testing sets, with 80% in the training set and 20% in the testing set.

1. Testing the Algorithms:

For each of the 5 models that we used, the model was trained on the training set and evaluated using the testing set using four common classification metrics:

Accuracy: the overall proportion of correct predictions

Precision: the ratio of true positives to predicted positives

Recall: the ratio of true positives to all actual positives

F1 Score: the mean of precision and recall

We used the models and algorithms imported from the sklearn library in order to call and evaluate each algorithm.

After running all 5 of the algorithms, the results are tabulated below:

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.771921	0.534968	0.769935	0.631297
Random Forest	0.837726	0.676038	0.691503	0.683683
Naive Bayes	0.815846	0.706811	0.467974	0.563114
Support Vector Machine	0.787005	0.551277	0.860784	0.672110
k-Nearest Neighbour	0.798110	0.574713	0.784314	0.663350

2. Analysis of Testing Results:

Testing Results:

The Logistic Regression model is a simple and easy to interpret model, with fairly high recall and F1 score, but accuracy and precision.

The Random Forest algorithm performed the highest in accuracy, precision, and F1 score, with a slightly lower recall than support vector machine and k-nearest neighbour.

The Naive Bayes model ran the fastest, but it does not perform well with feature independence, causing it to have the lowest recall and F1 score out of all the algorithms that were tested.

Support Vector Machine is slow to train and sensitive to feature scaling, but this algorithm performed decently well across all results, with the highest recall score.

The k-Nearest Neighbour model also performed reasonably well, but scored the lowest precision score out of all the algorithms and may be sensitive to irrelevant features and noise.

Analysis:

From these results, we concluded that the Random Forest model does the best job overall at predicting annual income using the dataset provided, and without feature selection. Logistic Regression and the Support Vector Machine model were closely behind in terms of accuracy and precision, while Naive Bayes and k-Nearest Neighbour had significant drawbacks that caused a reduction in recall and precision, respectively.

These results highlight the value of testing multiple algorithms when performing analysis on a dataset, as performance between models can vary significantly depending on the characteristics of each dataset. In the next task, we will incorporate a feature selection algorithm in order to further assess the impact of dimensionality reduction on model accuracy.

Task 3 - Feature Selection Algorithms

In order to improve the model interpretability, as well as potentially enhance performance, we decided to apply a feature selection algorithm to the dataset. The dataset contains a variety of information in each record such as demographic or employment characteristics, where some may be more important than others. By applying a feature selection algorithm, we may be able to reduce noise, lower the risk of overfitting (which could be an issue when applying SMOTE) and see which features had the most impact on our outcome.

We decided on two algorithms for feature selection, random forest's built in feature importance, as well as SelectKBest method using mutual information. These were chosen as they rank the contribution of each feature and are useful for our specific dataset where there are both numerical and categorical features.

Ranking of Features Based On Random Forest

Top 10 Features (Random Forest Importance)	
Feature	**Importance (Measured in %)**
age	0.2170
education-num	0.1378
relationship	0.1308
capital-gain	0.1253
hours-per-week	0.1080
occupation	0.0851
marital-status	0.0551
workclass	0.0490
capital-loss	0.0383

Top 10 Features (Random Forest Importance)	
native-country	0.0194

Ranking of Features Based on SelectKBest Mutual Info

Top 10 Features (SelectKBest - Mutual Info)	
Feature	**Importance (Measured in %)**
relationship	0.1187
marital-status	0.1102
capital-gain	0.0846
age	0.0710
education-num	0.0675
occupation	0.0620
hours-per-week	0.0409
capital-loss	0.0360
sex	0.0291
workclass	0.0123

Analysis of Findings

Some notable variables that the Random Forest model identified were education-num, capital-gain, hours-per-week, and marital status. These attributes make sense as they reflect how one's education, investment income, and work effort can greatly affect one's income. SelectKBest also shows the importance of capital-gain, age, education-num, and more, further validating the importance of these features. These findings validate our expectations and give us a smaller subset of features that we can use to potentially build a more efficient and interpretable model.

Task 4 - Comparison of Algorithms With and Without Feature Selection

In order to make the comparison between data mining accuracy between algorithms that use feature selection and algorithms without feature selection, as well as the importance of the features, we applied the top 3 features found in Task 3 to the original 5 algorithms used in Task 2. The top 3 features found using the Random Forest Model were: age, education-num, and relationship.

1. Testing the Algorithms With Feature Selection:

First, the top 3 features (age, education-num, and relationship) were selected, and all other columns (features) were dropped. As before, the features were scaled, split into 80/20 training/testing data, and SMOTE was applied for class balancing. All 5 models were evaluated with this new dataset, with the results shown below:

Accuracy Comparison Chart		
Model	**Accuracy (without feature selection)**	**Accuracy (with feature selection)**
Logistic Regression	0.771921	0.710260
Random Forest	0.837063	0.766120
Naive Bayes	0.815846	0.710758
Support Vector Machine	0.787005	0.760816
k-Nearest Neighbour	0.798110	0.793801

Precision Comparison Chart		
Model	**Precision (without feature selection)**	**Precision (with feature selection)**

Precision Comparison Chart		
Logistic Regression	0.534968	0.454545
Random Forest	0.673872	0.525836
Naive Bayes	0.706811	0.458638
Support Vector Machine	0.551277	0.517214
k-Nearest Neighbour	0.574713	0.608994

Recall Comparison Chart		
Model	**Recall (without feature selection)**	**Recall (with feature selection)**
Logistic Regression	0.769935	0.712418
Random Forest	0.692810	0.791503
Naive Bayes	0.467974	0.779085
Support Vector Machine	0.860784	0.854248
k-Nearest Neighbour	0.784314	0.522222

F1 Score Comparison Chart		
Model	**F1 Score (without feature selection)**	**F1 Score (with feature selection)**
Logistic Regression	0.631297	0.554990
Random Forest	0.683210	0.631881
Naive Bayes	0.563114	0.577380
Support Vector Machine	0.672110	0.644318
k-Nearest Neighbour	0.663350	0.562280

2. Comparison to Results Without Feature Selection:

With the exception of the Naive Bayes model, all 4 other models performed similarly with the top 3 features selected when compared to having all features selected. As mentioned in Task 2, the Naive Bayes model does not perform well with feature independence, so it is expected that the results improved with feature selection, as shown in the results. Support Vector Machine experienced almost no degradation in the results in all categories, suggesting that the top 3 features captured most of the relevant data necessary for class prediction. Logistic Regression showed some slight sensitivity to the reduction in features, in line with the algorithm's reliance on feature distribution. k-Nearest Neighbour showed a noticeable reduction in recall, with slight increases to precision and no change in accuracy. Notably, k-NN is also the only model to show an increase in precision when feature selection is applied, with all other models showing a reduction in precision. The Random Forest model increased recall, but lost accuracy and precision. When feature selection is applied, Support Vector Machine has pulled ahead of the Random Forest model (without feature selection) in doing the best job overall at predicting annual income using the dataset provided with feature selection.

In conclusion, the feature selection phase highlighted that a small subset of features, those related to age, education, and relationship, carry significant predictive power for income classification. Given the near-equivalent performance of 4 out of the 5 models trained only on the top 3 features, and improved performance in the Naive Bayes model, when compared to models trained with all features selected, feature selection led to simpler models with fewer inputs, leading to faster training times, improved interpretability, and lower risk of overfitting with only a minor trade off in performance.

Conclusion and Closing Statements

Throughout this project we explored the Adult Census Income Dataset, applying a series of preprocessing steps, class balancing techniques, algorithm comparisons, and feature selection algorithms to build an effective model for predicting if an adult made above or below 50k per year. The results of our data preprocessing showed that our cleaning and class balancing significantly improved model performance, particularly in terms of recall and F1 score.

Among the five algorithms we test, Random Forest produced the strongest and most consistent results across the metrics of accuracy, precision, recall, and F1 score, before applying feature selection.

Our feature selection algorithm helped to reduce the complexity of our model without significantly compromising the effectiveness of the model, further improving model performance under certain conditions. After feature selection the five algorithms were re-tested and we found the SVM algorithm now had the strongest results.

Through our preprocessing and analysis we were able to successfully develop and evaluate a model using the Random Forest and SVM algorithms, which were able to provide an accurate classification of an individual's income level based on census data, both with and without specialized feature selection.