Report

Classification model

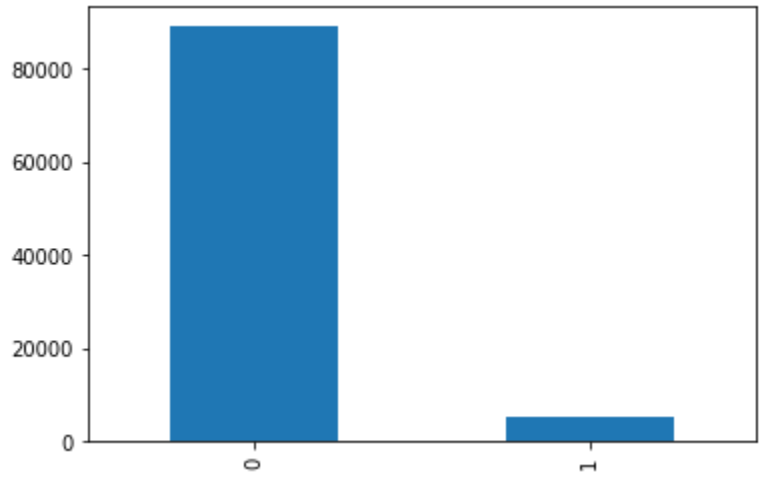
Data Cleaning

- The dataset contained null values, both numerical and categorical values. The categorical values were both nominal and ordinal. The data had redundant columns as well.
- Since the missing values were represented by '?' , they were replaced by NAN values and removed after detection.
- The dependent column, 'income' which is to be predicted has been replaced with 0 and 1 and hence convert the problem to a dichotomous classification problem.
- The unnecessary data points and redundant attributes have been removed, it is necessary to select the set of attributes really contributing to the prediction of the income.

Data pre-processing:

- To check the correlation between a binary variable and continuous variables, the point biserial correlation has been used.
- I have observed that 'instance weight' has almost 0 correlation. So I have removed this from the dataset and continued the analysis.
  Note: I have used instance weight for class weight calculation which I have explained in the following section.
- First, the categorical variables are encoded or rather dummies are generated and the numerical values are normalized to be between [0,1]. It's simply a case of getting all your data on the same scale: if the scales for different features are wildly different, this can have a knock-on effect on your ability to learn (depending on what methods you're using to do it). Ensuring standardized feature values implicitly weights all features equally in their representation.

For Imbalanced Dataset:



Clearly, the dataset is imbalanced on class 0. So I tried 2 standard techniques: Resampling & Class Weights.

Resampling: I have accomplished using SMOTE analysis.
Class Weights: I have manually calculated weights using a hypothesis based on assumptions given in the question. The question states that instance weight for each sample is a relative proportion of that instance in the whole population. Using this, I have intuitively calculated that

$$w_{class} = \frac{(sum(instance\ weight) \times total\ sample)}{sum(instance\ weight\ for\ that\ class) \times total\ sample\ for\ that\ class}$$

Then I modeled my classifier with this manually calculated weights.

Model architecture:

The training and testing is divided in 70–30 for logistic, naive bayes whereas decision tree and random forest.

Used standard sklearn models

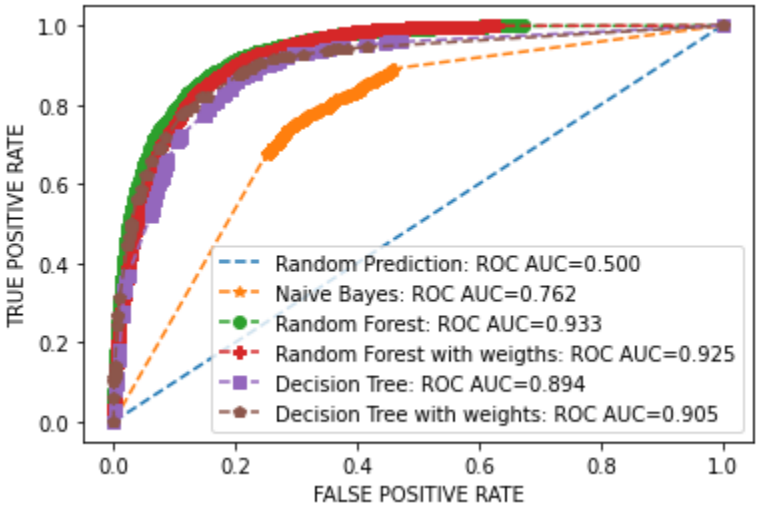
- **Logistic regression**
  - Used both class weights and resampled dataset and created two models
- **Naive bayes Classifier**
  - Used cross entropy criterion
- **Decision Tree Classifier**
  - Used both class weights and resampled dataset and created two models
  - Used standard identifiers of depth and samples

- **Random Forest Classifier**
    - Used both class weights and resampled dataset and created two models
    - Used standard identifiers of samples

These are the standard classification ensemble and linear methods for such data. Since the number of data points is not very high, such models perform better and scale well. In such cases, ideally such ensembles are chosen because of their explainability and scalability.
Neural networks and deep models like ANN also can give good accuracy but such models tend to overfit and do not perform well on test data.

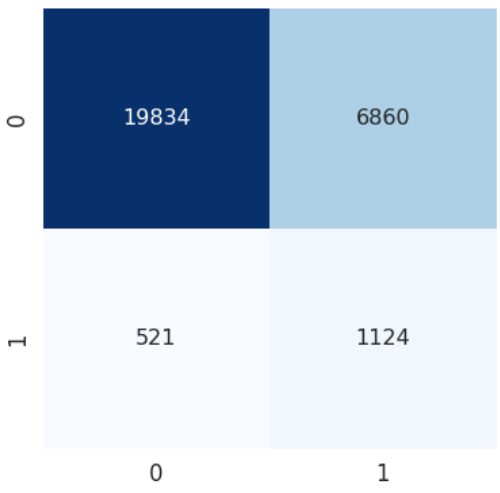
Training algorithm & evaluation

ROC Curve

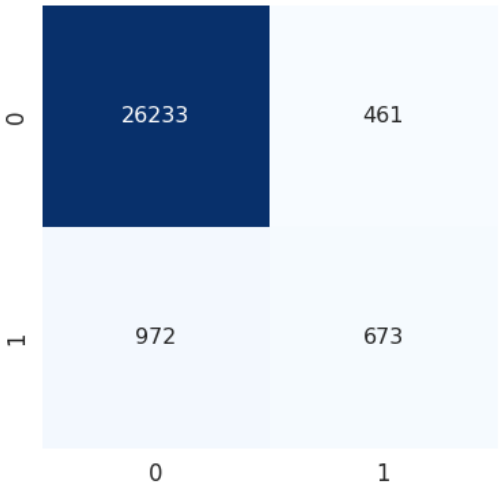


Based on this Random Forest performs best

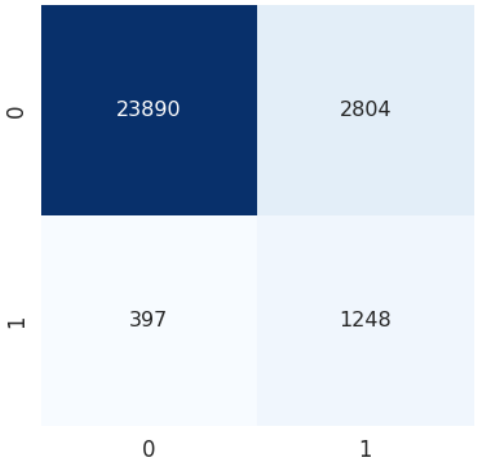
Confusion Matrix:



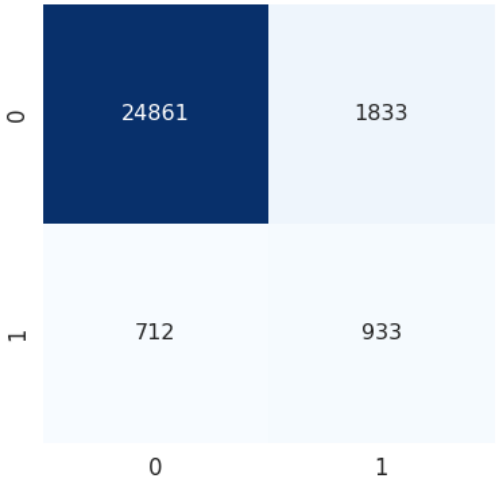
Naive Bayes



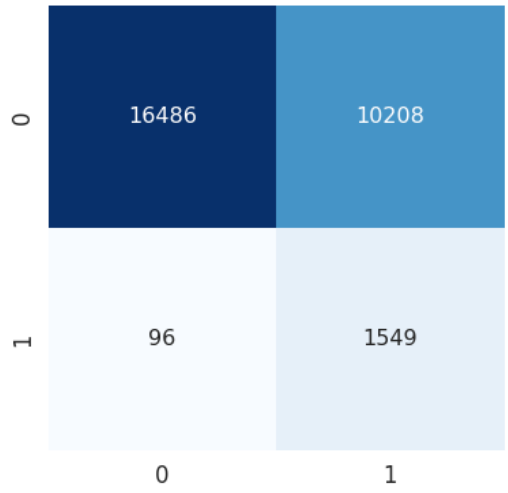
Random Forest



Random Forest with Class Weights



Decision Tree



Decision Tree with Class Weights

Based on this analysis, Random Forest with class Weights performed best

Segmentation Model

- **Data Preprocessing:**
  - After pruning all the redundant columns the data columns I had left with that I decided as the feature space for rudimentary segmentation model was as follows.
  - I have encoded every class in feature space and decreased the encoded classes in few features such as education level, marital status.

    Eg: I combined '1st - 12th grade' into a single encoding rather than 4 individual encodings as per the raw data. This decreases the feature space for clustering methodology to work on basic features. Several such methods were used to decrease the categories.

Feature Description

- **Categorical Attributes**
  - class of worker
  - detailed industry recode
  - detailed occupation recode
  - education
  - marital status

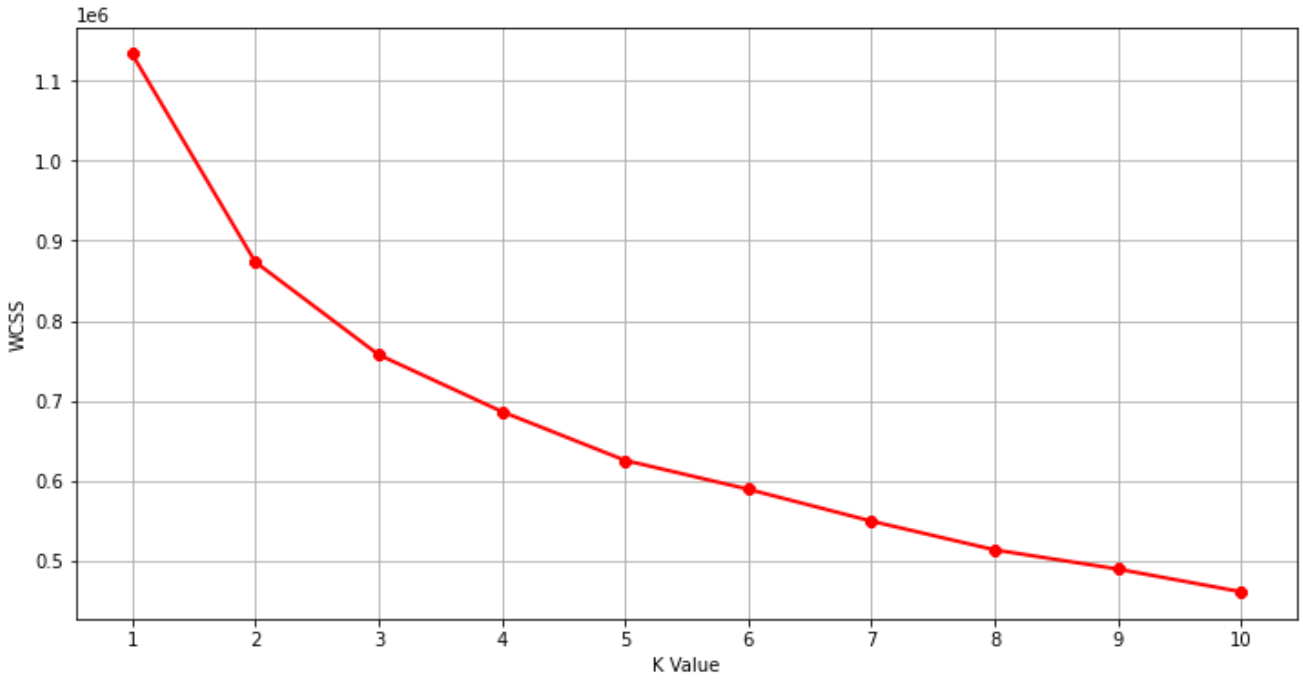
- race
- sex
- tax filer stat
- state of previous residence
- detailed household summary in household
- num persons worked for employer
- citizenship

- **Model Architecture**

  There are many machine learning algorithms, each suitable for a specific type of problem. One very common machine learning algorithm that's suitable for customer segmentation problems is the k-means clustering algorithm.

  Implementation:

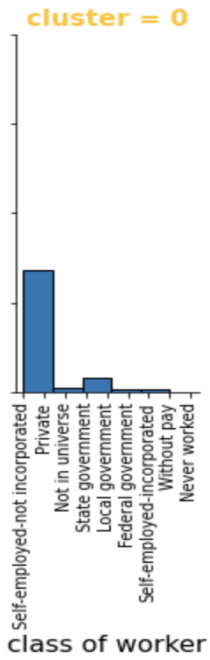
  - Performed Elbow Method to determine the number of clusters.

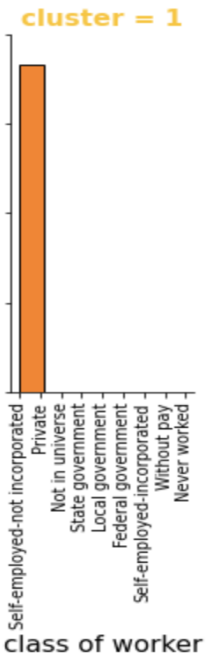


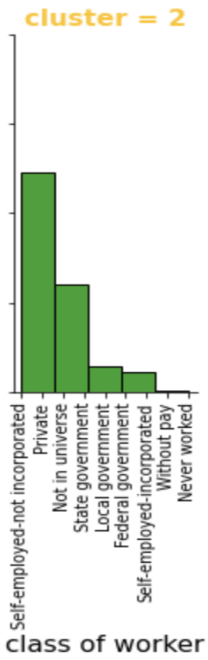
Based

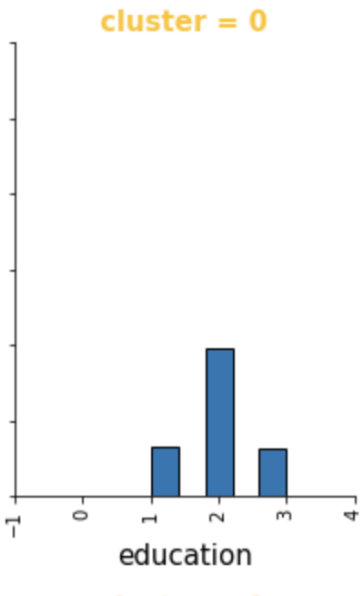
Clusters:

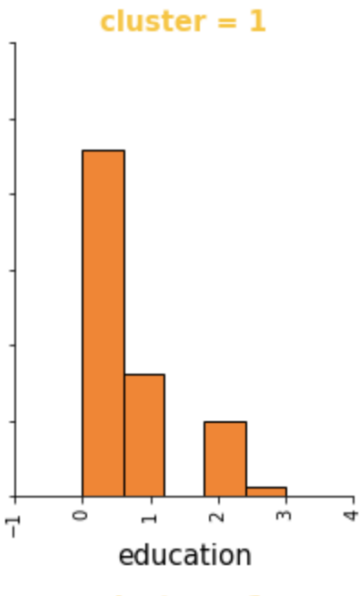
Major takeaways after clustering:

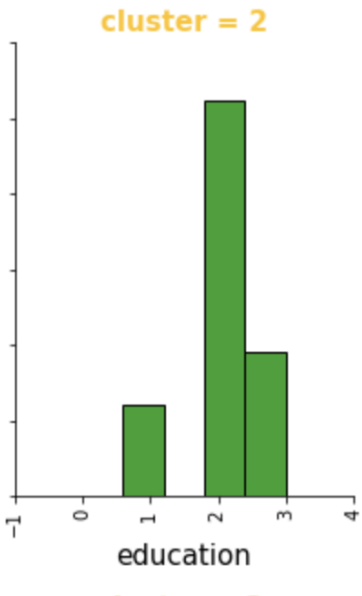


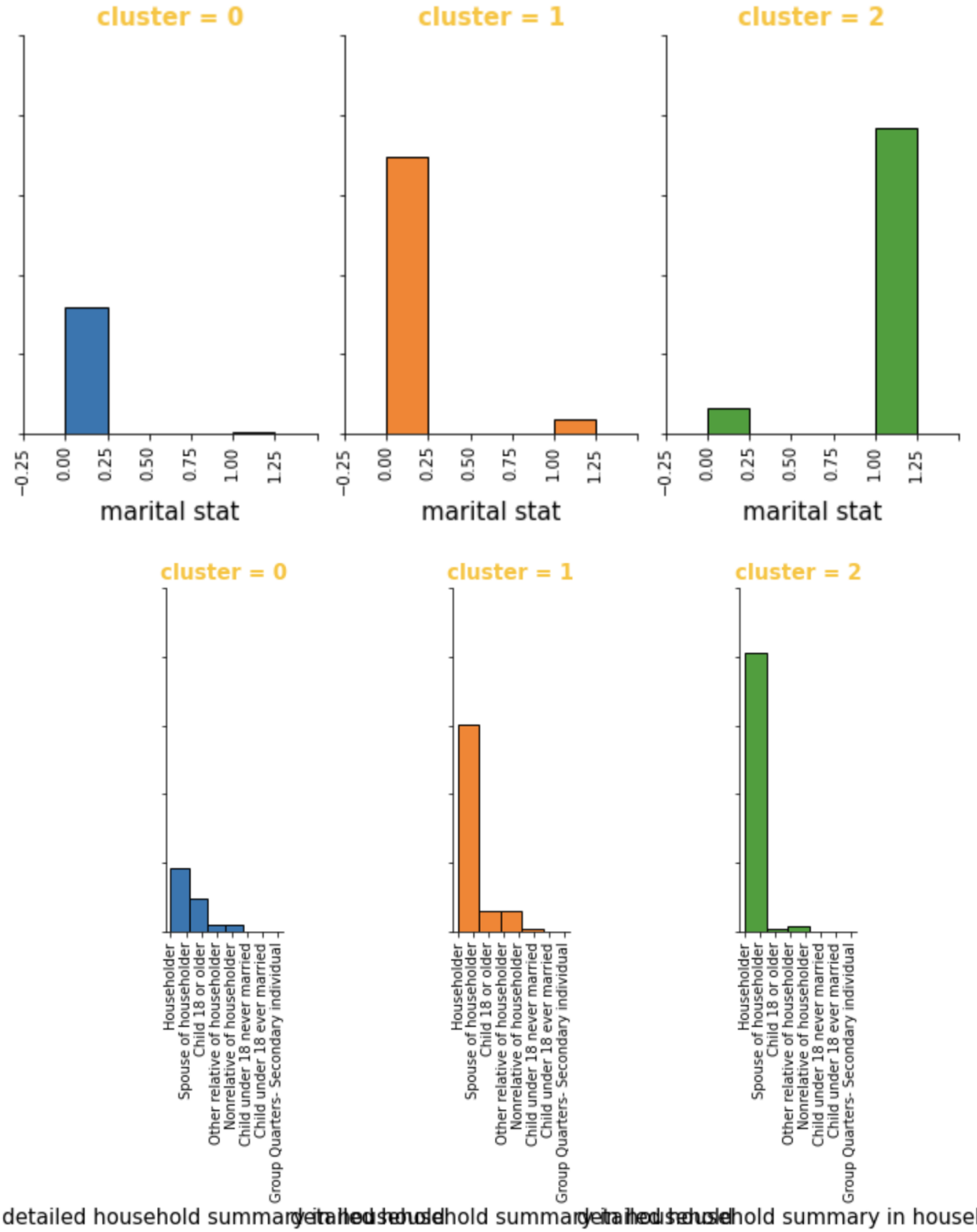


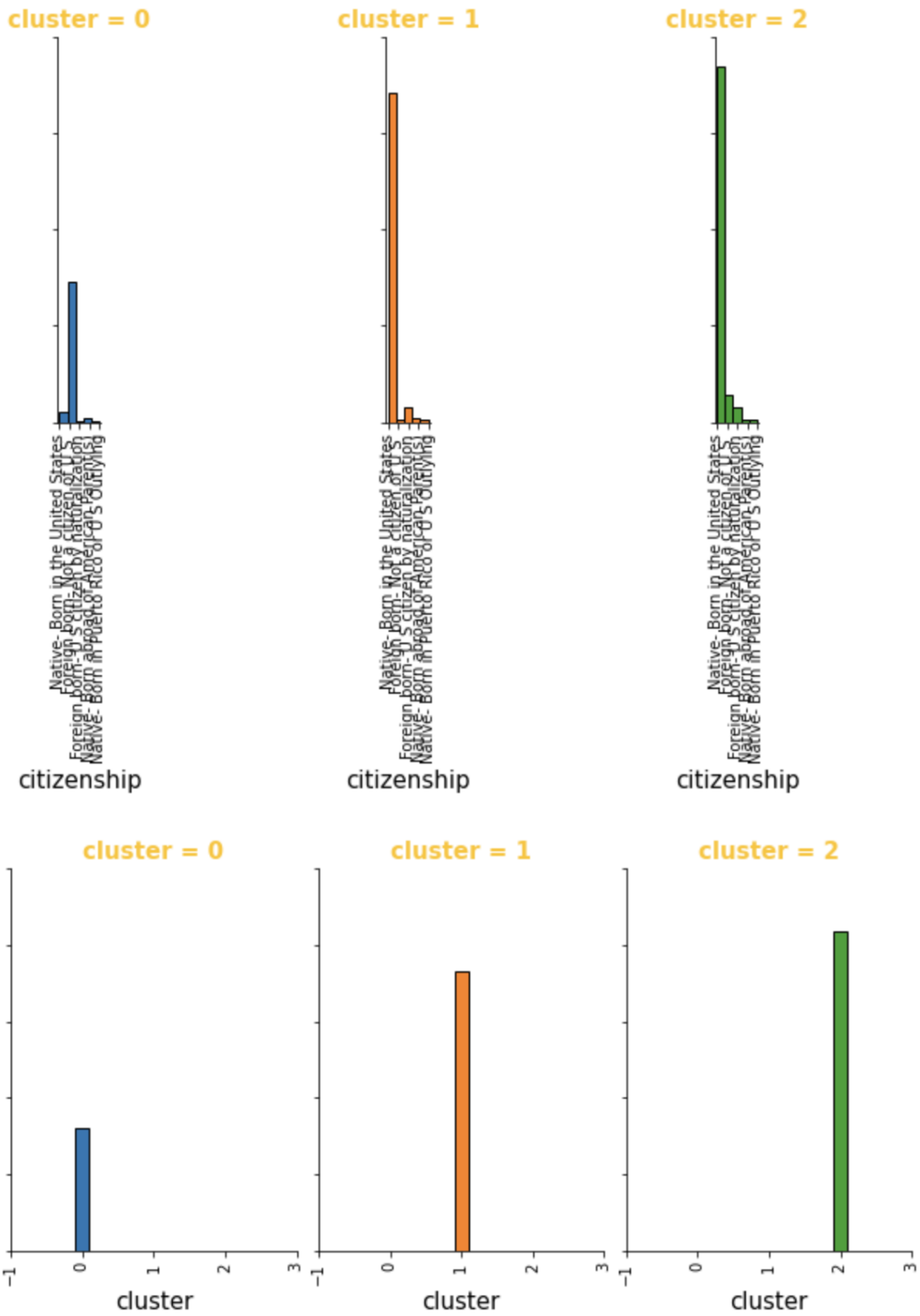









cluster = 0
cluster = 1
cluster = 2
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
marital stat
Householder
Spouse of householder
Child 18 or older
Other relative of householder
Nonrelative of householder
Child under 18 never married
Child under 18 ever married
Group Quarters- Secondary individual
detailed household summary in household


cluster = 0
cluster = 1
cluster = 2
Native- Born in the United States
Foreign-born- Not a citizen of U.S.
Foreign-born- U.S. citizen by naturalization
Native- Born abroad of American parent(s)
citizenship
cluster
−1
0
1
2
3

Cluster stats with income level:

- ***Cluster 0 : People with >50000$: 863***
  - ***Indicates good chance of high income***
- ***Cluster 1 : People with >50000$: 113***
  - ***Indicated low chance of high income***
- ***Cluster 2 : People with >50000$: 4466***
  - ***Indicated very high chance of high income***

Based on this my rudimentary segmentation model is as follows:
Criteria for each cluster are decided based on the above graphs of clusters.

Cluster 2: High probability of high income:

Criteria: Highly educated(Masters or above), Self-employed/Private employed, Married, Household owner, Native US Citizen

Cluster 0: Moderate probability of high income:

Criteria: Educated(Bachelors or College level), Self-employed/Private employed, Unmarried, Non- Household owner, Foreign National (predominant)

Cluster 1: Low probability of high income:

Criteria: Uneducated/High school Graduates, Unmarried, Native US Citizens, Unemployed/Self employed

The above would be my rudimentary segmentation model for this data.

References:

1. https://arxiv.org/pdf/1810.10076.pdf
2. https://www.analyticsvidhya.com/blog/2016/02/guide-build-predictive-models-segmentation/
3. https://neptune.ai/blog/customer-segmentation-using-machine-learning