**Census Income Data Set**

**Introduction:**

This is an exercise, performed by Mohamed and Jacob, to use machine learning to solve a classification problem. The dataset in which we will be performing analysis/prediction can be found here, on the UCI Machine Learning Repository site: <http://archive.ics.uci.edu/ml/datasets/Census+Income>. This particular data set, compiled from census data, contains 14 attributes (See below) defining a person’s demographic, behavioral and socioeconomic characteristics in an attempt to predict income levels- above or below a threshold of $50k per year. The data set is robust, comparatively speaking, containing 48,842 instances. Also included is a test dataset that we can crossvalidate our models against.

\*All code used to generate graphs and data cleansing is included in the RMD file.

**Attribute Definitions:**

Attributes are either categorical or integers, and some attributes contain missing values within both datasets (training and test)

Class: >50K, <=50K.

**age**: continuous.

**workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

**fnlwgt:** continuous.

**education**: 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**: continuous.

**marital-status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

o**ccupation**: 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**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

**race**: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

**sex**: Female, Male.

**capital-gain**: continuous.

**capital-loss**: continuous.

**hours-per-week**: continuous.

**native-country**: 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.

Of these attributes, fnlwgt was indecipherable to a person without Census domain knowledge. It is an abbreviation for finalweight, and acts as a measurement of similarity between two unique individuals, regarding economic and behavioral characteristics. It should be noted that fnlwgt is a relevant metrics, if and only if it is used to compare individuals within the same state- “People with similar demographic characteristics should have similar weights. There is one important caveat to remember about this statement. That is that since the CPS sample is actually a collection of 51 state samples, each with its own probability of selection, the statement only applies within state” (http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names).

**Initial Strategies:**

There are a wide range of techniques that we can employ to build our models using both R and Weka (We will use both to contrast performance). First, we will use **Association Rules Mining** to see if any particular attributes are highly correlated with income (Setting the RHS to Income), and evaluate performance using support, confidence and lift. This kind of analysis would be useful if someone were using Census data to target specific individuals, but was unsure of their income level- For instance, we could see that a male with a doctoral degree is likely to have an income exceeding 50k, or Income YES.

We will utilize **Decision Tree** theory for help to predict classification per the training data set (induction/deduction). Our root nodes will be our attributes, as defined above, and our leaf nodes will be our decisions. In class we have focused heavily on the **J48 algorithm**, but we may also experiment with the **C4.5** tree. Through this analysis, we will derive entropy and **information gain/gain ratio** as well as **Gini and Error** for our attributes, in order to measure importance (Similar to P Values in a regression problem). When building decision tree models, we will employ vast parameter refinements/tuning (Confidence Factor, minNumObj, Reduced Error Pruning, Binary Split, Subtree Raising) in an attempt to create the model with the best performance. Our model evaluation will focus on accuracy, precision, recall and f measure as we perform a variation of holdout tests and N Fold Cross Validation models and compare results.

**Our goal** will be to have a model that performs better than both random guess (50/50) and majority vote (N/48,842). We will document our results via excel and note any changes made to models. We will also experiment with adjustments to our random seeds and take the means/standard deviations of our accuracy per those adjustments.

We will also implement measures of Naive Bayes, or Bayesian Theory, in an attempt to solve our classification problem. Naive Bayes focuses on instance probability, and co-occurrence, or conditional, probability. We will look at the variable of Class as an event that is dependent on our other variables. Initial munging is incomplete at this point, so we will need to better understand our continuous variables and decide if smoothing is needed or not. Results will be compiled and compared in a similar manner to our decision tree findings.

Bayes' theorem is a mathematical formula for determining conditional probability. To calculate the probability of B given A, the algorithm counts the number of cases where A and B occur together and divides it by the number of cases where A occurs alone.

Prob(B given A) = Prob(A and B)/Prob(A)

Applying to our data set, our goal is to explore **Naive Bayes** to calculate the probability of B given A, the algorithm counts the number of cases where A and B occur together and divides it by the number of cases where A occurs alone.

We are aiming to detect if Naive-Bayesian classifiers are robust to irrelevant attributes, and classification takes into account evidence from many attributes to make the final prediction. From previous experiments conducted using NB classifiers, we have seen that supervised discretization provides enhanced accuracy levels, and this will be tested throughout this process.

Covered later will be SVMs, Random Forests and KNN models to evaluate and compare key metrics amongst the mentioned classifiers.

**Experiment Preprocessing**

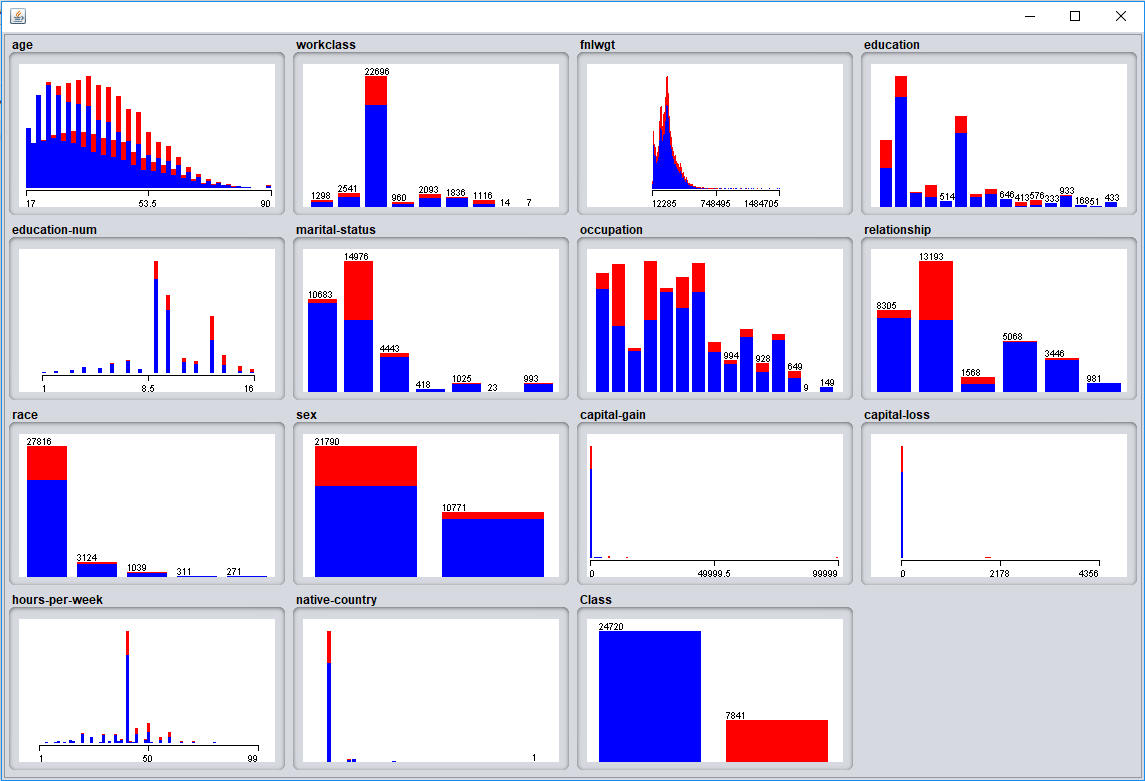
Our preprocessing steps will vary per the data mining method and data questions that we are trying to answer. Exact measures of preprocessing will be noted per the method/algorithm being used, however main preprocessing takes place in R. The Census data must be read from the web, and since Weka does not have those capabilities, we start in R.

**Summary Statistics**

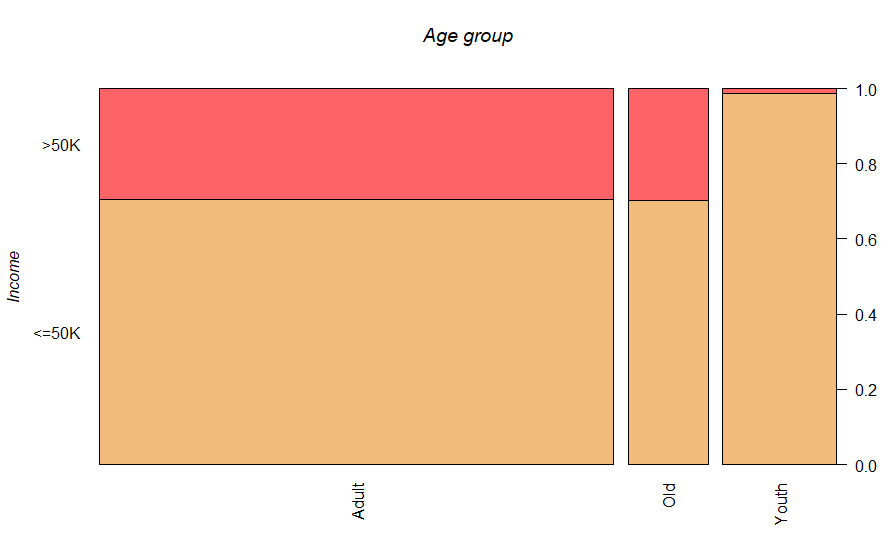
Initial summary statistics performed via R and Weka will show us distribution by variable, and help us to understand any issues we may have with multicollinearity or heteroskedasticity.

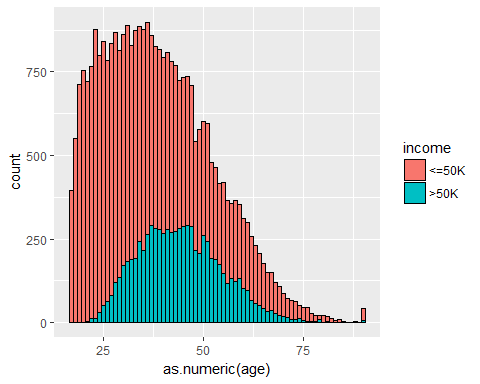
*Something we will touch upon with our classification models will be our accuracy rating. A random guess model could run at 50% accuracy, and a majority vote model could run at, in this dataset, 75% (22654 instances of class>50k out of 30162 total instances.*

*Basic distribution amongst features:*

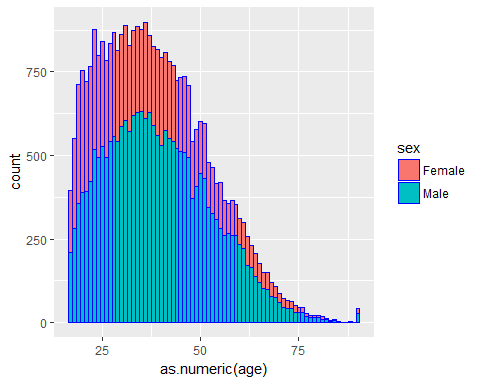
****

*Age in relation to class:*

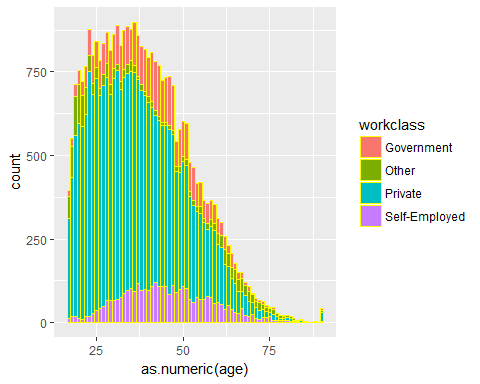
****



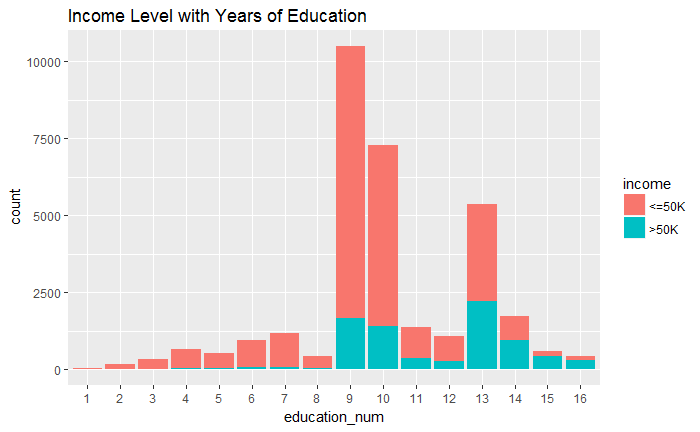
*Age in relation to Gender distribution:*



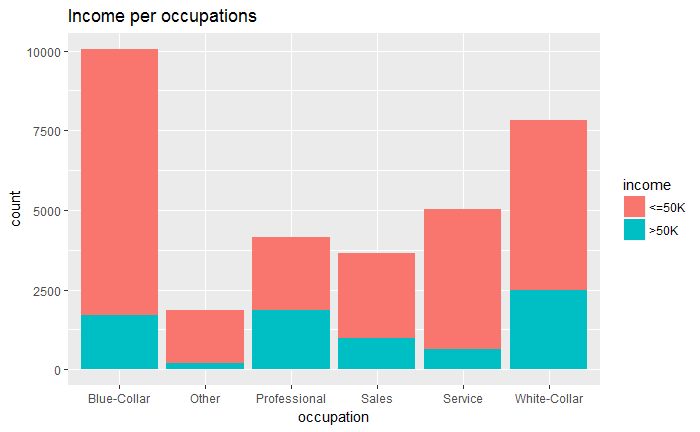
*Age in relation to Workclass:*



There does appear do be a correlation between the probability of an individual making greater than $50k per year, and the level of education they have reached/completed:



Also, white collar professions have the highest count of individuals making over $50k per year within our dataset, respectively:



**Hypothesis**

We believe that there are certain features, or variables, that are more influential on a person’s prospective income levels than others. Without running any tests, and just using domain knowledge to assert assumptions, we generally lean towards education being the most significant attribute in this dataset. The higher level of education completed seems to have a direct relationship with earnings potential. Because this isn’t a straightforward regression problem where our dependent variable is continuous and we are trying to measure the relationship between the variance of dependent and independent variables (ie. is the change in x responsible for the change in y), we will instead rely on the aforementioned techniques to conduct both supervised and unsupervised machine learning in an attempt to enlighten nontrivial factors in wealth.

Moving towards the classification models, we will engage in measures of parameter tuning while documenting accuracy and other important metrics per set of algorithmic rules. Random Forests showed promise with other datasets, and we are interested to see if Ensemble Learning provides a substantiated difference in accuracy with the Census data.

**Association Rules Mining and Cluster Analysis**

One of the many data mining techniques that we will explore is association rules mining using the Apriori algorithm. Per the algorithm, our tuning will be heavily reliant on Support and Confidence thresholds, but we will also explore the data utilizing the lift metric. A quick recap on AR Model metrics:

**Support:** Fraction of transactions that contain both x and y support.

**Confidence:** How frequently items in Y appear in transactions containing X.

**Lift:** A measure of expectancy that adjusts for transactions that are not as common, or are highly common- This helps us to understand if the occurrence of X had a positive or negative effect on Y.

It should be noted that these metrics have a way of measuring relationships between instances, but that relationship ends at correlation. Correlation does not equate to causation.

This is not a typical basket analysis as the data is not in transactional- The below will examine if AR Models work just as well on record data.

Now to start with rule generation, but first, we remember that the Apriori algorithm does not work for continuous variables. We venture back to the Preprocessing tab and use the Discretize filter, first-last, with the number of bins set to 10 as per default.

The following scheme details parameter tuning per our first iteration of rule generation:

**weka.associations.Apriori -N 50 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -A -c -1**

Using a support of 0.3 and a confidence level of .7 with our right hand side set to the class variable, we can begin to unveil some patterns:

**education-num='(8.5-10]' capital-gain='(-inf-9999.9]' capital-loss='(-inf-435.6]' 16923 ==> class=<=50K 14291 conf:(0.84)**

**education-num='(8.5-10]' race=White capital-gain='(-inf-9999.9]' capital-loss='(-inf-435.6]' 14345 ==> class=<=50K 11933 conf:(0.83)**

**workclass=Private capital-gain='(-inf-9999.9]' capital-loss='(-inf-435.6]' 21264 ==> class=<=50K 17227 conf:(0.81)**

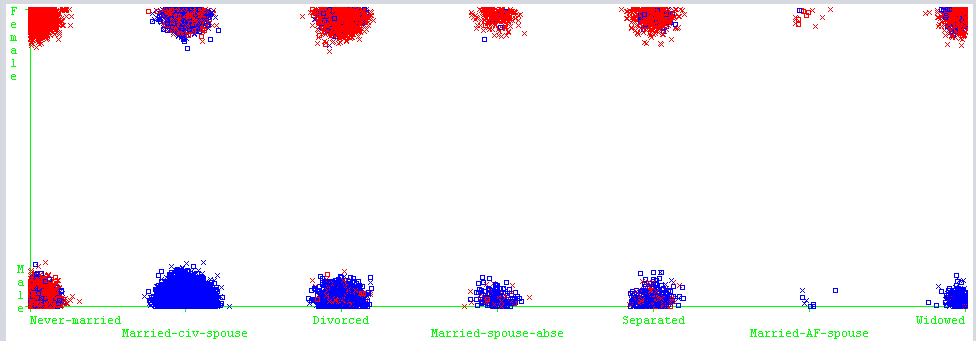
**hours-per-week='(30.4-40.2]' 17735 ==> class=<=50K 14103 conf:(0.8)**

The above rules note a relationship between education, capital gain, capital loss, race and hours worked per week. Looking at capital gain and capital loss first, we can see a highly skewed distribution towards the left side, meaning that most people, at a clip of around 96%, fall within the first bin. Using inference and glancing back at our summary statistics, we discover that most people are not investors, so this isn’t particularly useful. Next we look at education, which is the most prominent feature within our ruleset of 50. Our 8th-10th bins of education-num reflects a grade completion level of 11.5-infiniti. This is useful. People with higher levels of completed education generally make more money. Looking at race and hours-per-week, we also see uneven distributions where a single bin contains at least 50% of total instances. We expect these bins to be present, so they are not statistically significant.

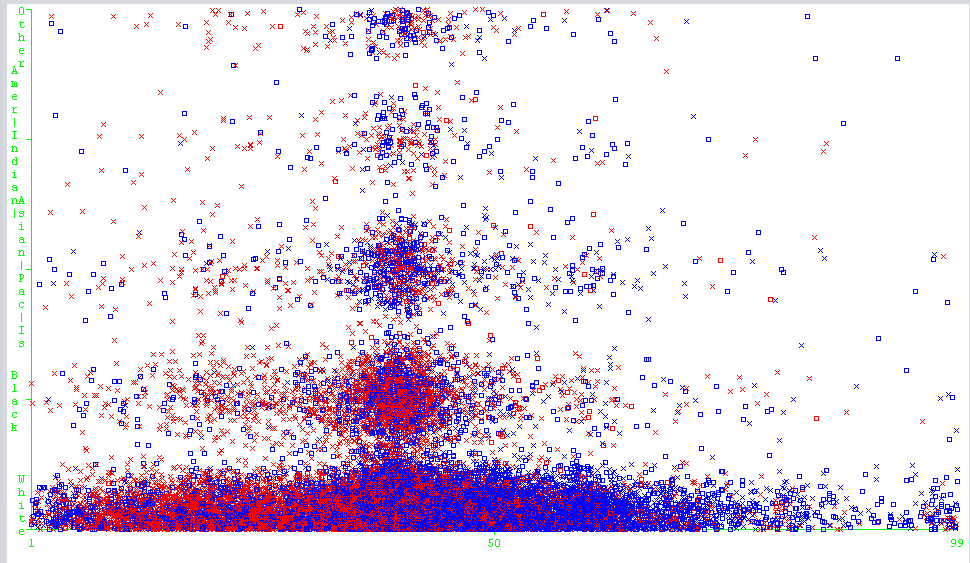
Clustering is a form of unsupervised learning and can be useful for exploratory purposes, or outlier detection. Using the Simple K Means algorithm along, and Euclidean distance\* as our Distance Function with numClusters = 2 (Cluster 0 relating to Class > 50k, and Cluster 1 relating to Class <50K) we see the following:

\* sqrt(sum((a-b)^2))

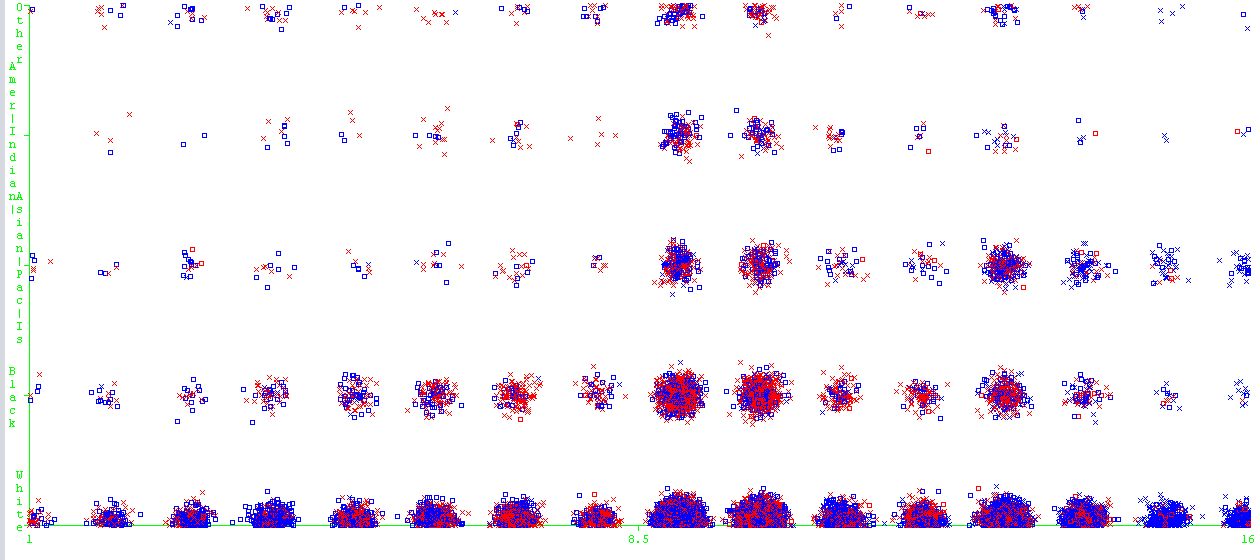
Males are more represented in our dataset, but interestingly, only Females who fall under the marital status of Married-Civ-Spouse have a concentrated focus of instances where class > 50k.



African Americans who work 40 hours a week are much more likely, probability wise, to fall within Cluster 1, as opposed to Caucasian people working the same amount and seeing a higher propensity to fall within cluster 0.

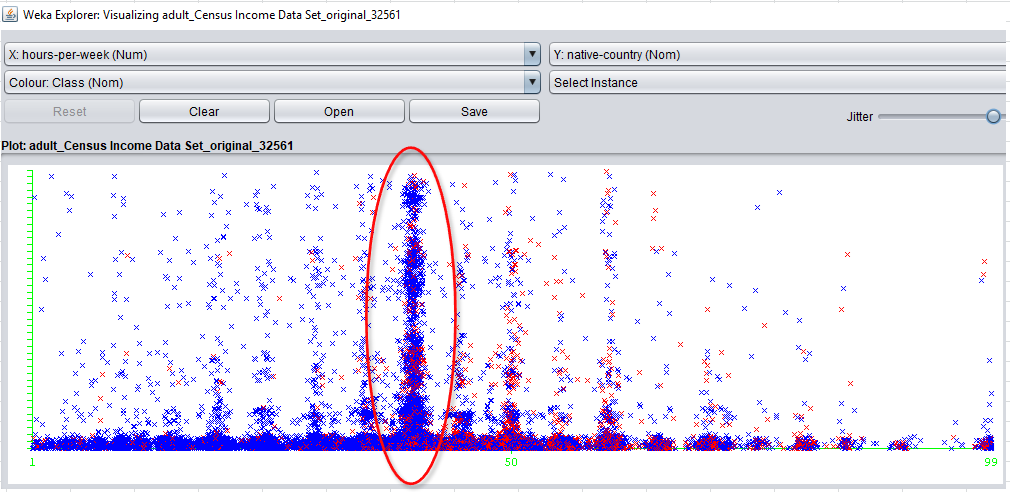


Looking at education-number versus race:

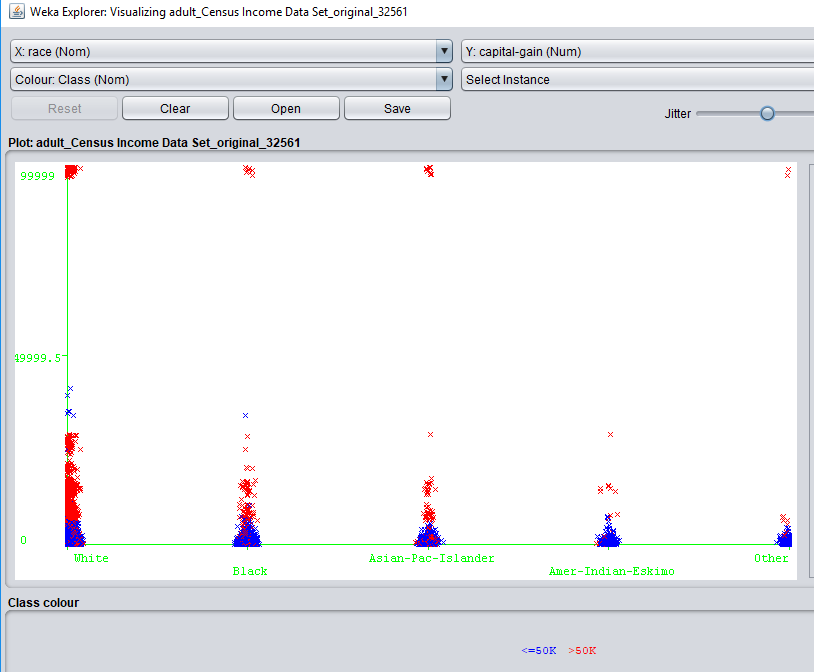


This is interesting. We know Caucasian people are heavily represented within the dataset, but seeing uneven distributions of income levels amongst race led us to look at education level in comparison to income. It appears that education-num doesn’t amplify corresponding instances as the scale increases, but it does show that when African Americans have the same level of education as Caucasians, there is a higher likelihood that they fall within the cluster of class < 50k.

There is a uniformed work scheduling protocol amongst the native countries noted within the Census data. 40 Hours per week is the most heavily populated using clustering techniques:

****

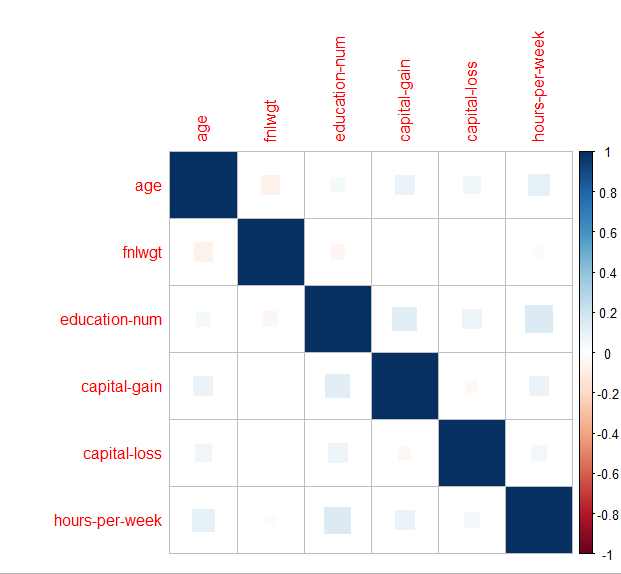
Looking at race vs capital gain, there’s a greater distribution of total capital gain going towards Caucasian individuals than any other race:

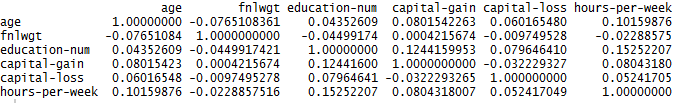
****

**Classification Models**

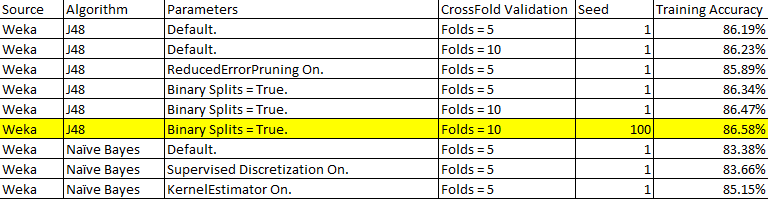
**Precursor:** After AR modeling, we had an inkling that Naive Bayes may not perform well due to high levels of multicollinearity amongst our independent variables, and the independence assumption that this algorithm operates with. Running a correlation test on the numeric features shows us that there are very miniscule relationships between any of the five noted below, so we will proceed with utilization of the Naive Bayes classifier in this experiment. We are also hesitant to run Random Forests due to their complexity/size and the robust nature of our dataset. Because there is no correct answer when choosing a model to perform induction, we will want to layer in variety and run as many different algorithms as possible and then deduce rationale after the fact to better understand our data in relation to the best model. With only 15 attributes present within our dataset, we lean towards the J48 algorithm providing a solid result due to the corresponding size of the trees, and the lessened complexity of the model.

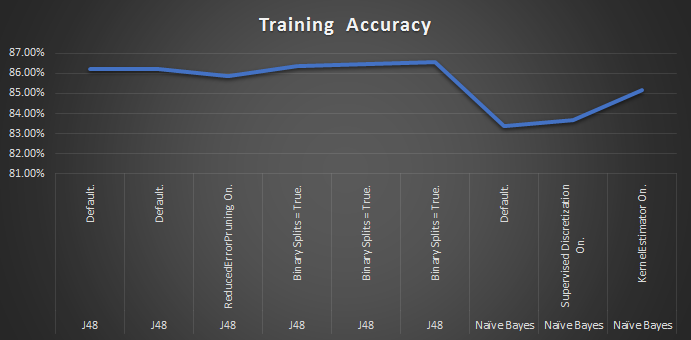
***Correlation Test Amongst Numeric Variables***

**

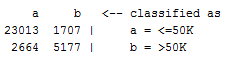
**

The below record data shows some of our results using a variety of classification models, as well as parameter tuning, on the Census data to predict classed based on the given attributes. It should be noted that the ID column was removed in preprocessing prior to induction:





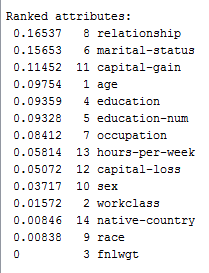
Starting with the less complex classification algorithms, we see the above results. Through parameterization, we were able to manufacture an accuracy rating of 86.58% using Crossfold Validation (Folds = 10) and a random seed set to 100. Here we discovered that the size of K in folds was influential on the results. A confusion matrix shows the following:



What this matrix tells us is that the key metrics, Precision/Recall/FMeasure, were at their lowest when dealing with class b, meaning there was a high rate of False Positives present within our prediction. As we’ve learned, a false positive is not statistically as bad as a false negative.

* In this case a false positive, in the bottom left of the matrix, occurs when the instance is class a, but it’s predicted as class b. In association with our data, this means that a person with income greater than 50k is predicted to have income less than 50k.
* If we look at the other side, a false negative occurs when an instance registers income greater than 50k, but our prediction mis-classifies them into bucket b- Bucket b being income less than 50k.

Our models are too large to visualize, but we can look at information gain of our best performing model to better understand which variables are statistically significant.

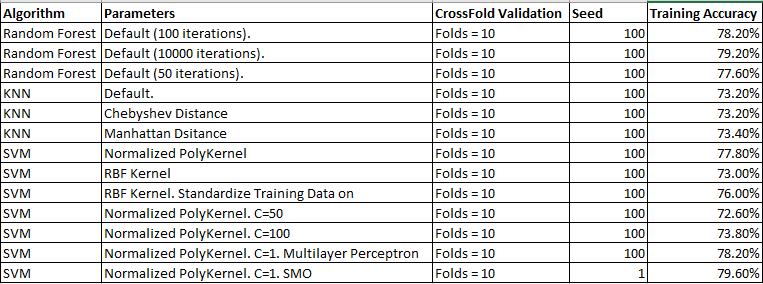


Interestingly enough, this IG counteracts what our cluster analysis told us, as we see Race and Sex as being less than influential on the model. Relationship and Marital Status are variables seen higher up in the nodes of the tree.

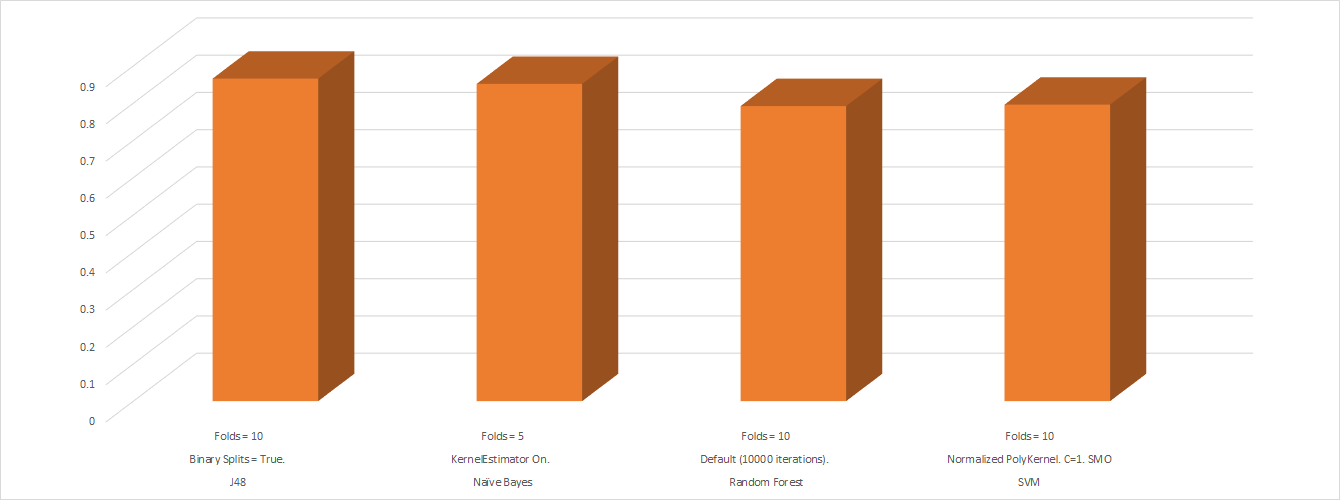
In order to perform induction using more complex models, such as SVM/KNN/Random Forests, or even ANN/CNN, we simply don’t have the computing power necessary to do so on a dataset containing 30k+ instances. What we must do at this point is systematically sample our dataset to reduce its overall size.

Our random sample of the dataset contains 73% instances classified as income less than 50k, so this is a sharp pivot from our original dataset and should thus be noted, as apples to apples control tests are thrown out the window. For the purposes of this assignment, we will divide our algorithm types between basic, and complex, with the following being the latter category.

* Support Vector Machines are one of the most popular machine learning algorithms around, and were at the forefront before the emergence of artificial and convoluted neural networks. One of the positives about SVMs is that they can use kernel trick to put their inputs/instances into a multidimensional space allowing for a hyperplane to be drawn that wasn’t possible in a two-dimensional space.
* RFs are an example of Ensemble Machine Learning, which means that a bunch of weak learning classifiers can come together and form a strong classifier. The decision trees, depending on the number of iterations, are randomly sampled from the training set with replacement. Majority vote is used to determine prediction results amongst the many trees at the end of the induction process. They are, however, nearly impossible to visualize due to their complexity.
* We know that KNN is considered a lazy algorithm – After further research, we discover that it is lazy because it doesn’t participate in generalization, but rather uses the entire training set during the deduction phase of modeling. What is great about KNN is that it is nonparametric, which means it doesn’t make/rely on assumption about a probability distribution.



All three types of models underperformed our expectations- We also tried discretizing all variables to see if that what have an impact on the performance, but were met with very similar results. A simple bar graph shows the top performing model per each algorithm below. Accuracy close to 80% was navigable through the use SVMs and Random Forests, but there was greater prediction error compared to J48 and Naive Bayes.



\*The best model had the following setup: J48. Folds = 10. Binary Split on. Seed = 100. Full training set used.

**Conclusion**

Much of our explanation takes place above, but here are some key points from this experiment:

* The J48 decision tree algorithm outperformed all other models trained, and did so with very little parameter tuning. When we merged the test dataset, using rbind in R, we used our model instances that were held out from the induction process. We registered an accuracy of 85.96%, which was higher than our accuracy with our training set. This eliminates major overfitting issues as being a problem. Our model was generalized enough to fit to different instances. When we ran the full dataset with using the holdout method and our parameters from our best Random Forest model, we saw accuracy shoot up to 85%. This was considerable higher than our previous measures using CV. Perhaps this is an instance of underfitting.
  1. It is a unique case, in our opinion, to see a simple Decision Tree algorithm thrive in comparison to an ensemble machine learning technique such as Random Forests, which take an iterative approach to DTs.
* Association Rules Mining for data exploration was not particularly useful in this case. In order for us to see even distribution amongst features we would have had to bend the data in preprocessing. The results noted above were so heavily skewed by Married, White Males that most rules contain some variation of that audience subset.
* Clusterers helped to eliminate the above issue, in a way, by showing us density amongst certain attributes. Our findings are noted above.

**Learnings/What Could Have Been Better**

* Systematic sampling, or sampling in general, would have been a more viable method to conduct induction testing- We adjusted towards the latter end of our experiment, but mitigated the impact of model comparisons. Most models generated very quickly, but Random Forests, KNNs and SVMs all took a few minutes to build. Because 30000 instances is not a large dataset in real world situations, we will account for model induction and deduction times moving forward
* Binning categorical variables into smaller buckets may have been beneficial to our model - For instance, three buckets for age, young/middle aged/old, rather than it being continuous.
* Removing the finalweight attribute due to lower impact on the model would have led to a cleaner dataset.
  1. We left it in because information gain told us that it had no impact at all. Also, with education and education number present, we could have removed either one of them and eliminated forced collinearity.
* It would have been better to have a majority vote more akin to random guess, meaning an even distribution amongst our class variable.
* It may have been more beneficial for us to conduct feature selection after running information gain to only focus on the variables that were important to our classification.