Census Data Classification Report

CPS844 - Data Mining

Professor Cherie Ding

Mike Francki (500554567)
Abel MacNeil (500563525)

The problem:
The prediction task is to determine whether a person makes over 50K a year.

Resources:
census data (1994 Census database)
32 561 instances

This census data was extracted from the census bureau database found at
http://www.census.gov/ftp/pub/DES/www/welcome.html

Files Available:
https://archive.ics.uci.edu/ml/machine-learning-databases/adult/

Creating the Arff File:
The data provided from the above link gave us the total data in a csv file, where every line represents an instance, and the attribute values for the instance are comma separated. The order and type of the attributes for the data were given in a separate file. To manipulate the given data into the weka format (.arff) we had to first take all the attribute information and remove the periods at the end of every line, and provide the appropriate annotations for Weka. Once we had the attributes in the arff file format we merely needed to paste the data from the csv file in the arff file after the attribute information. We placed “@data” before the data as this corresponds to the arff file format.

Attributes used for prediction (14; 6 real and 8 Nominal) :

Age: REAL

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

fnlwgt: REAL

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: REAL

Marital-status: {Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse}

Occupation: {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: REAL

Capital-loss: REAL

Hours-per-week: REAL

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}

Notes from the Authors of the Data:
Conversion of original data as follows:
1. Discretized agrossincome into two ranges with threshold 50,000.
2. Convert U.S. to US to avoid periods.
3. Convert Unknown to "?"
4. Run MLC++ GenCVFiles to generate data,test.

Description of fnlwgt (final weight)

The weights on the CPS files are controlled to independent estimates of the civilian noninstitutional population of the US. These are prepared monthly for us by Population Division here at the Census Bureau. We use 3 sets of controls.
These are:

1. A single cell estimate of the population 16+ for each state.
2. Controls for Hispanic Origin by age and sex.
3. Controls by Race, age and sex.

We use all three sets of controls in our weighting program and "rake" through them 6 times so that by the end we come back to all the controls we used.

The term estimate refers to population totals derived from CPS by creating "weighted tallies" of any specified socio-economic characteristics of the population.

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.

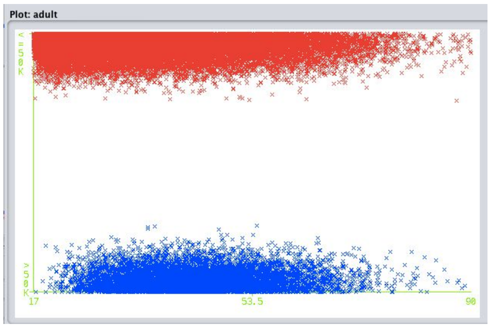
Class to predict (Binary)

Yearly income
Class 1: >50k
Class 2: <=50k

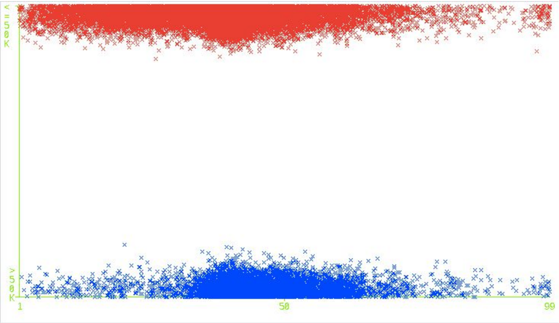
Understanding The Data

The first step is to get a better understanding of the data by generating graphs to visualize the relationship between the attributes and the class.

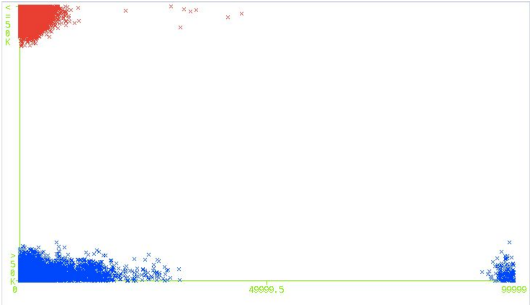
Age



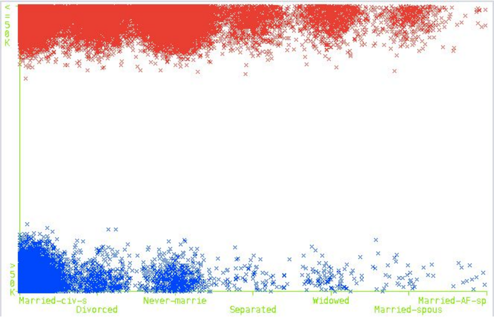
Hours per week



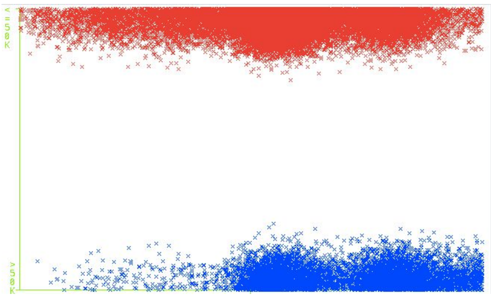
Capital-gain



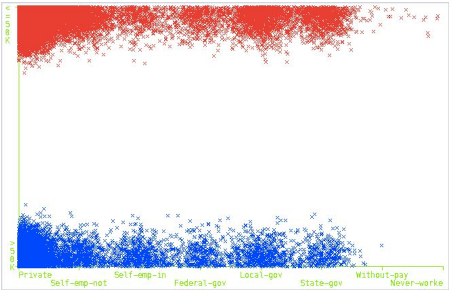
Marital-status



Education-num



Workclass



General Observations

People making >50K
Mid aged
Working full time hours
Higher Capital gain
Married
More years of education
Work in the private sector
People making <=50K
For the most part the relations had less structure and patterns making prediction harder
Less capital gain
Fewer hours worked
Less education

Attributes

In this section we determine which attributes are most useful for instance classification, and whether or not attribute selection helps in making accurate predictions.

Attribute	Average Merit	Average Rank
fnlwgt	0.797 +- 0.002	1.1 +- 0.3
age	0.813 +- 0.001	2.1 +- 0.3
hours-per-week	0.831 +- 0.001	3.1 +- 0.3
occupation	0.833 +- 0.025	5 +- 3
workclass	0.847 +- 0.002	5.6 +- 1.5
native-country	0.85 +- 0.001	6.6 +- 0.66
marital-status	0.851 +- 0.004	7.1 +- 0.94
race	0.852 +- 0.004	7.2 +- 0.87
sex	0.856 +- 0.001	9 +- 0
education	0.85 +- 0.018	9.1 +- 2.7
capital-loss	0.848 +- 0.001	11 +- 0
capital-gain	0.82 +- 0.001	12 +- 0
relationship	0.778 +- 0.006	13 +- 0
education-num	0.767 +- 0.024	13.1 +- 2.7

This table showcases the merit of each attribute. The attributes that had the least contribution to classifying the data are listed at the top, the most valuable attributes are at the bottom. To retrieve this data we ran the WrapperSubsetEval in Weka, using the IBk (k=1) algorithm with 10 cross-validation folds. It is from this table we can determine the best attribute subsets (the subset of n attributes that yields the greatest accuracy), by repeatedly running the IBk (k=1, cross-validation 10) algorithm after removing the worst attribute.

Number of Attributes	**Attributes**	**Correctly Classified Instances**
14	fnlwgt, age, hours-per-week, occupation, workclass, native-country, marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	79.4171 %
13	age, hours-per-week, occupation, workclass, native-country, marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	79.9085 %
12	hours-per-week, occupation, workclass, native-country, marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	81.5976 %
11	occupation, workclass, native-country, marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	83.4802 %
10	workclass, native-country, marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	84.2327 %
9	native-country, marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	84.6995 %
8	marital-status, race, sex, education, capital-loss, capital-gain, relationship, education-num	84.9237 %
7	race, sex, education, capital-loss, capital-gain, relationship, education-num	85.3014 %
6	sex, education, capital-loss, capital-gain, relationship, education-num	85.5809 %
5	education, capital-loss, capital-gain, relationship, education-num	85.7683 %
4	capital-loss, capital-gain, relationship, education-num	85.7897 %

3	capital-gain, relationship, education-num	84.7978 %
2	relationship, education-num	81.9262 %
1	education-num	77.9583 %
0	None (just class)	75.9190 %

According to these results the best subset of attributes to predict the individual’s salary class, is the 4th subset {capital-loss, capital-gain, relationship, education-num}. Thus it is these 4 attributes that are most important when trying to classify new instances. It is evident that attribute selection provides a much greater prediction accuracy; there is a 6.37% improvement in the percentage of correctly classified instances from classifying with all the attributes, compared to only using the best 4 attributes.

Models

zeroR: all attributes

Test Options	**Class Predicted**	**Correctly Classified Instances**
Cross-validation(10)	<=50k	75.919 %
Cross-validation(5)	<=50K	75.919 %
Percentage split %66	<=50K	76.5062 %
Percentage split %80	<=50K	76.7199 %
Use training set	<=50K	75.919 %

Using the simplest classifier zeroR, we observe that majority class is <=50k. Based on this training set we should predict about 76% of people make less or equal to 50k

OneR: all attributes

Test Options	**Classifier model**	**Correctly Classified**
Cross-validation (10) Min bucket Size 6	capital-gain: < 3048.0 -> <=50K < 3120.0 -> >50K < 4243.5 -> <=50K < 4401.0 -> >50K < 4668.5 -> <=50K < 4826.0 -> >50K < 4932.5 -> <=50K < 4973.5 -> >50K < 5119.0 -> <=50K < 5316.5 -> >50K < 5505.5 -> <=50K < 6457.5 -> >50K < 7073.5 -> <=50K < 10543.0 -> >50K < 10585.5 -> <=50K < 30961.5 -> >50K < 70654.5 -> <=50K >= 70654.5 -> >50K	80.9066 %
Cross-validation (10) Min bucket Size 8	capital-gain: < 3048.0 -> <=50K < 3120.0 -> >50K < 4243.5 -> <=50K < 4401.0 -> >50K < 4668.5 -> <=50K < 4826.0 -> >50K < 5119.0 -> <=50K < 5316.5 -> >50K < 5505.5 -> <=50K < 6457.5 -> >50K < 7073.5 -> <=50K >= 7073.5 -> >50K	80.8913 %
Cross-validation (10) Min bucket Size 12	capital-gain: < 3048.0 -> <=50K < 3120.0 -> >50K < 4243.5 -> <=50K < 4401.0 -> >50K < 4668.5 -> <=50K < 4826.0 -> >50K < 5119.0 -> <=50K < 5316.5 -> >50K	80.8698 %

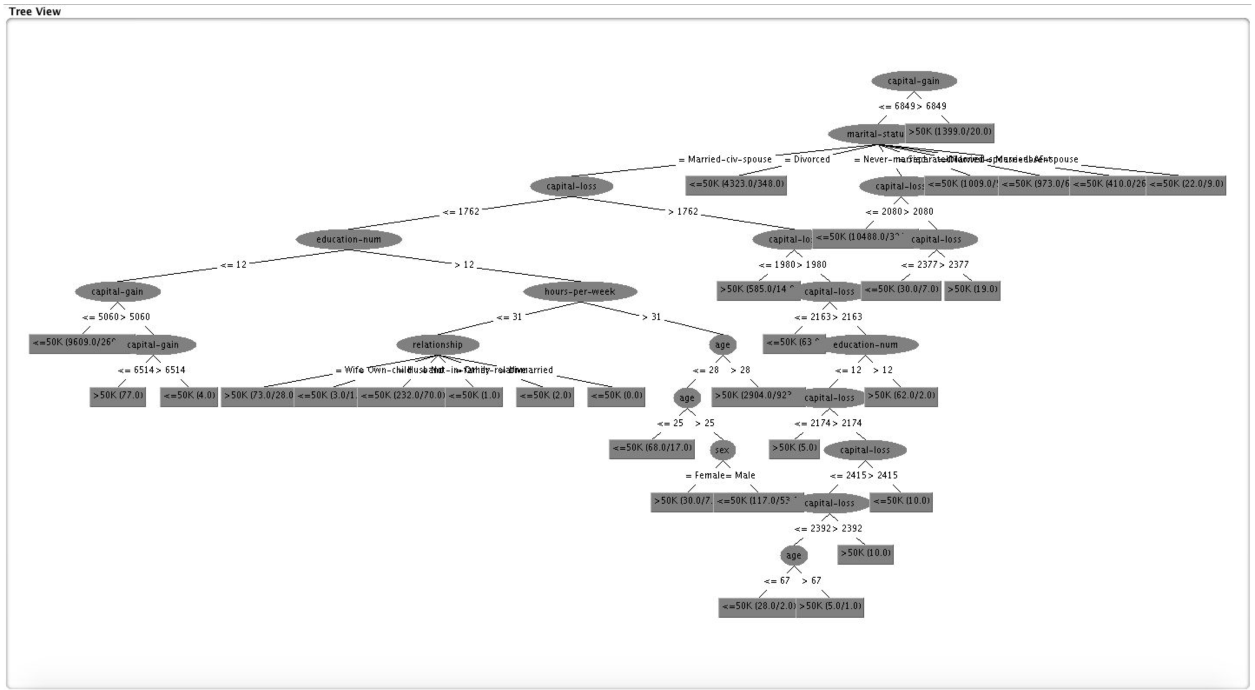
	< 7073.5 -> <=50K >= 7073.5 -> >50K	
Cross-validation (10) Min bucket Size 35	capital-gain: < 3048.0 -> <=50K < 3120.0 -> >50K < 4243.5 -> <=50K < 4401.0 -> >50K < 5119.0 -> <=50K < 5316.5 -> >50K < 7073.5 -> <=50K >= 7073.5 -> >50K	80.7868 %
Cross-validation (10) Min bucket Size 60 Cross-validation (10) Min bucket Size 70	 capital-gain: < 3048.0 -> <=50K < 3120.0 -> >50K < 5119.0 -> <=50K < 5316.5 -> >50K < 7073.5 -> <=50K >= 7073.5 -> >50K capital-gain: < 3048.0 -> <=50K < 3120.0 -> >50K < 5119.0 -> <=50K >= 5119.0 -> >50K	80.5166 % 80.4644 %

OneR selected capital-gain as the most important attribute for decision making. Capital gain was and attribute in the best subset for attribute selection {capital-loss, capital-gain, relationship, education-num}. Originally ran with a max bucket size of 6 for discretizing the attribute this gave a highly flexible model with 17 ranges this seems to obviously overfit by inspecting the the graph visualizing capital gain vs class. We increased the bucket size to 100, a much less flexible model with only 2 ranges, but this had almost no effect on the accuracy of predictions. This model should be used following principle of simplicity, as it is more likely to give better predictions to new instances.

J48: All attributes

Test Options	**Number of leaves**	**Size of tree**	**Correctly Classified**
cross -validation(10) Confidence Factor 0.25	564	710	86.2105 %
cross -validation(10) Confidence Factor 0.20	377	484	86.2381 %
cross -validation(10) Confidence Factor 0.15	317	406	86.3241 %
cross -validation(10) Confidence Factor 0.10	205	270	86.1767 %
cross -validation(10) Confidence Factor 0.05	105	150	85.9187 %
cross -validation(10) Confidence Factor 0.025	30	50	85.4949 %
cross -validation(10) Confidence Factor 0.01	30	50	85.4581 %
cross -validation(10) Confidence Factor 0.005	30	50	85.4366 %

I decreased the Confidence Factor to increase the pruning of the tree to generate a less flexible model to decrease overfitting. Decreasing it below 0.025 seemed to have no effect in the pruning and the accuracy went down slightly. This is a better model to use for classifying new instances. It is less flexible and therefore less fitted to the training data and more so to the true underlying model.



Observation

When capital gain is larger than 6849 then predict >50k

Otherwise you go to inspect the other attributes to predict

J48: {capital-loss, capital-gain, relationship, education-num}

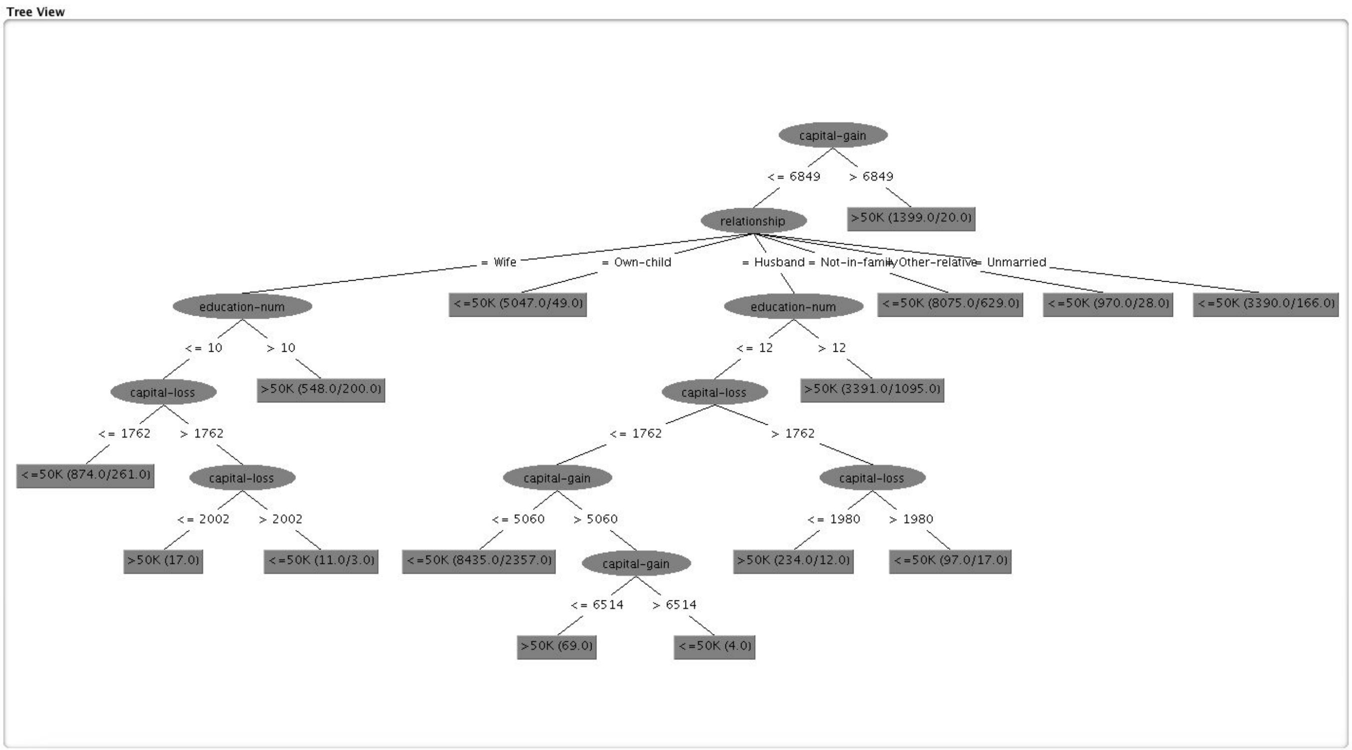
Test Options	Number of leaves	Size of tree	Correctly Classified
cross -validation(10) Confidence Factor 0.25	49	93	85.5318 %
cross -validation(10) Confidence Factor 0.20	49	93	85.5226 %
cross -validation(10) Confidence Factor 0.10	33	61	85.4765 %
cross -validation(10) Confidence Factor 0.05	28	51	85.4243 %
cross -validation(10) Confidence Factor 0.002	19	33	85.1755 %
cross -validation(10) Confidence Factor 0.001	15	25	85.0772 %
cross -validation(10) Confidence Factor 0.0005	15	25	85.0772 %

Fitting J48 with the subset of attributes {capital-loss, capital-gain, relationship, education-num} taht we determined to be the best subset there is almost no decrease in accuracy but the tree generated is much simpler. After adjusting the confidence factor to 0.001 to further prune the tree, it is very reasonably sized and easy to understand why tree was generated.

Comparison
attributes: all
Confidence Factor: 0.25
Size of tree: 710
accuracy : 86.2105 %

attributes: all
Confidence Factor: 0.001
Size of tree: 24
accuracy : 85.0772 %

There is ~2700% decrease of the size of tree
Only ~1% decrease in accuracy
This should should lead to less overfitting and better predictions of new instances



Best Model Generated with J48

- Using attribute selection and Increased pruning
- Similar to the tree using all attributes if capital-gain > 6849 predict >50K
- Otherwise inspect the other attributes to predict but the height of the subtree when capital-gain <= 6849 is now about half.
- This model is simpler and predicts approximately just as accurate.

Naive Bayes:

Test Options	**Correctly Classified Instances**
Cross-validation (10) All attributes	83.428 %
Cross-validation (5) All attributes	83.382 %
Cross-validation (10) {capital-loss, capital-gain, relationship, education-num}	79.7795 %
Cross-validation (5) {capital-loss, capital-gain, relationship, education-num}	79.7457 %

Attribute selection does not seem to help Naive bayes, the prediction become worse.

SimpleLogistic: All attributes

Test Options	Confusion Matrix	Correctly Classified
Cross validation (10)	a b <-- classified as 4655 3186 | a = >50K 1679 23041 | b = <=50K	85.0588 %
Cross validation (5)	a b <-- classified as 4660 3181 | a = >50K 1673 23047 | b = <=50K	85.0926 %
Percentage Split % 66	a b <-- classified as 1589 1012 | a = >50K 601 7869 | b = <=50K	85.4304 %
Percentage Split % 80	a b <-- classified as 904 612 | a = >50K 338 4658 | b = <=50K	85.4115 %
Use training data	a b <-- classified as 4656 3185 | a = >50K 1662 23058 | b = <=50K	85.1141 %

SimpleLogistic: {capital-loss, capital-gain, relationship, education-num}

Test Options	Confusion Matrix	Correctly Classified
Cross-validation (10)	a b <-- classified as 4354 3487 | a = >50K 1679 23041 | b = <=50K	84.1344 %
Cross-validation (5)	a b <-- classified as 4387 3454 | a = >50K 1710 23010 | b = <=50K	84.1405 %
Percentage Split %66	a b <-- classified as 1516 1085 | a = >50K 608 7862 | b = <=50K	84.7078 %
Percentage Split %80	a b <-- classified as 866 650 | a = >50K 356 4640 | b = <=50K	84.5516 %
Use Training set	a b <-- classified as 4359 3482 | a = >50K 1661 23059 | b = <=50K	84.205 %

Using attribute selection only slightly decreases accuracy therefore it is most likely best to use it because is create a less complex, less flexible model and therefore less prone to overfitting.

Observing the the confusion matrix is clear that is much easier to predict when the class is <=50K , ~93% accurate but when the class is >50K it is only ~60% accurate.

The J48 trees agree with these observations, as well as simple visualizations of the attributes vs class graphs.

Conclusion

ZeroR was surprising accurate (~76%) this suggests that a larger proportion of the true population make <=50k a year. Based on the other models tested, a more flexible model can improve these results.

OneR increased the accuracy (~81%), but the default setting generated a model with many intervals on the attribute. This suggested overfitting. By tuning the parameters we generated a model with only one split, however, much less flexible and therefore more resilient to over overfitting. This tuning left accuracy almost unchanged (~80.5). Attribute selection would not help here because of the type of model.

Using the IB1 algorithm proved to have worse results (79.4%) initially than OneR, but when choosing the best attributes, using the best 4-attribute subset the accuracy jumped to 85.8%. In this algorithm removing certain attributes clearly helped improve the performance of the IB1 algorithm.

J48 added more flexibility to the model increasing the model accuracy (~86%) but again with default settings it generated a huge tree, making it impossible to be displayed properly. Using both attribute selection and tuning of the pruning parameter, we generated a much smaller tree (less flexible) that was easier to display and understand. The accuracy was almost unchanged (~85%) due to this tuning.

Naive Bayes performed similarly but seems to be worse about 83% accuracy and attribute selection seemed to be a detriment to performance. More test data would help to compare further it still has potential to compete with the other models.

SimpleLogistic seemed to be approximately on par with J48 with about 85% accuracy without attribute selection and 84% with attribute selection.

It seems that it is easy to predict when the instance belongs to <=50K and more difficult when it belongs to >50K.

Best Model
J48 and SimpleLogistic appear to be the best models with similar accuracies, however with attribute selection IB1 provided almost the same performance. J48 generates a decision tree that is easy for humans to visualize and understand what the model is doing. This is potentially very useful, however, more testing is needed to determine the the most optimal pruning parameter. SimpleLogistic does not need the tuning of a parameter but also is more difficult to generate a human understandable visualization of the model.

Visualization Important
J48

Visualization not Important
SimpleLogistic