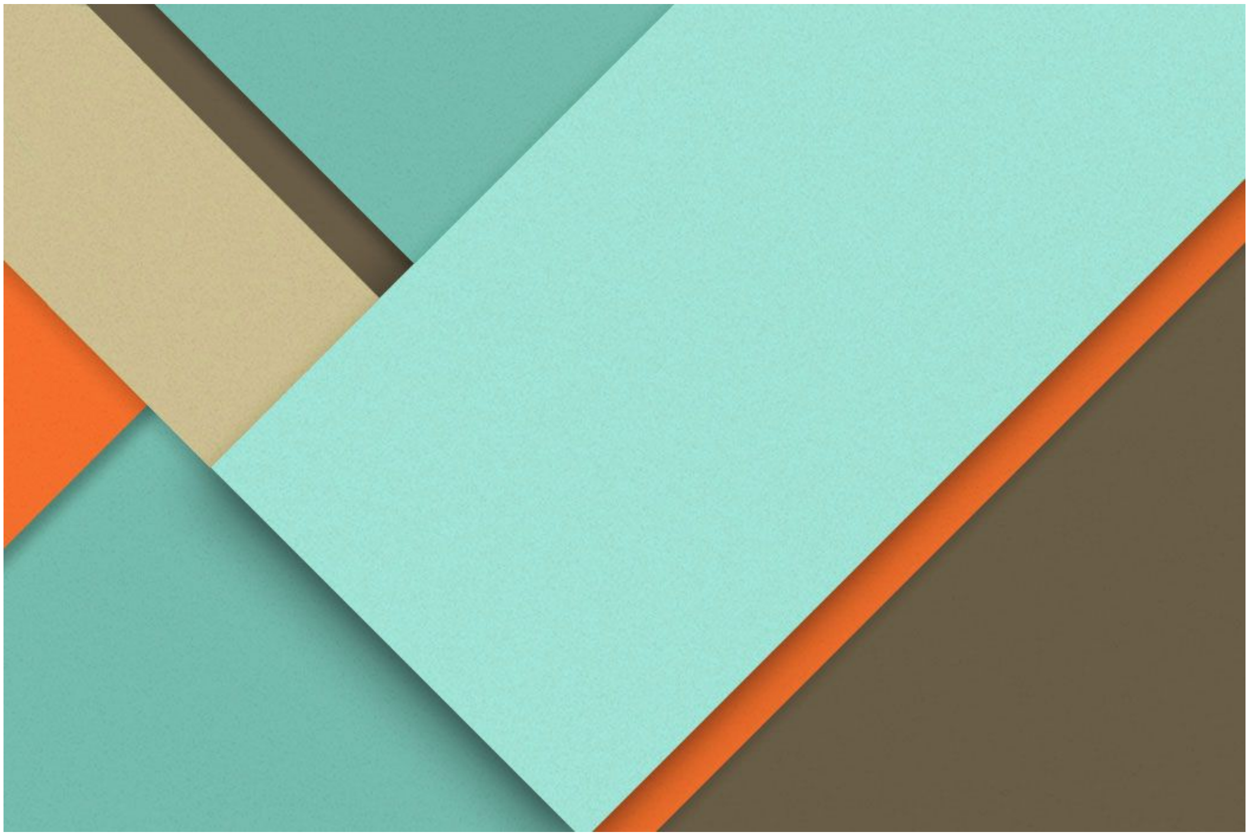


# Advanced Data Analytics: Assignment 2

11.10.2016

# Exploring the Data

The data was extracted from the 1994 Census Bureau database based in the United States of America. The class target aims to predict whether an individual's yearly income exceeds US$50k per annum.

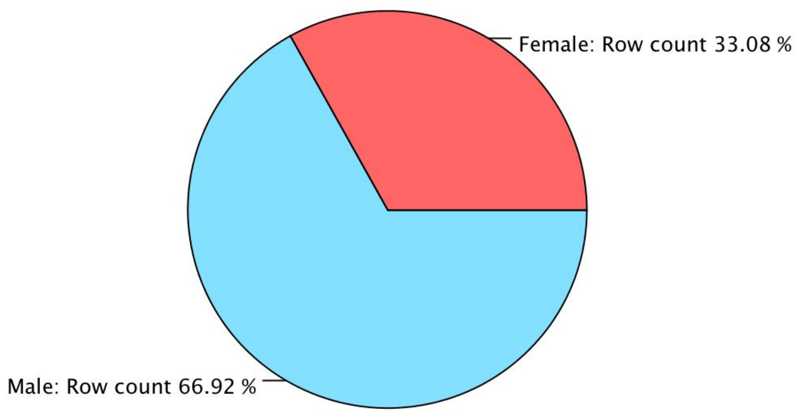
The data set contains 32 561 cases with the attributes:

| Name | Key | Description | Type |
|---|---|---|---|
| income (Target Class) | >50k, <=50k | Whether an individual earns a yearly income > or <= US$50k | nominal |
| age | Continuous | Age of individual | ratio |
| workclass | Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked | | nominal |
| fnlwgt | Continuous | | ratio |
| 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 | | nominal |
| education.num | Continuous | | ordinal |
| marital.status | Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse | | nominal |
| 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 | | nominal |
| relationship | Relationship status | | nominal |
| race | White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black | | nominal |
| sex | Female, Male | | nominal |
| capital.gain | Continuous | | ratio |
| capital.loss | Continuous | | ratio |
| hours.per.week | Continuous | | ratio |
| 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 | | nominal |

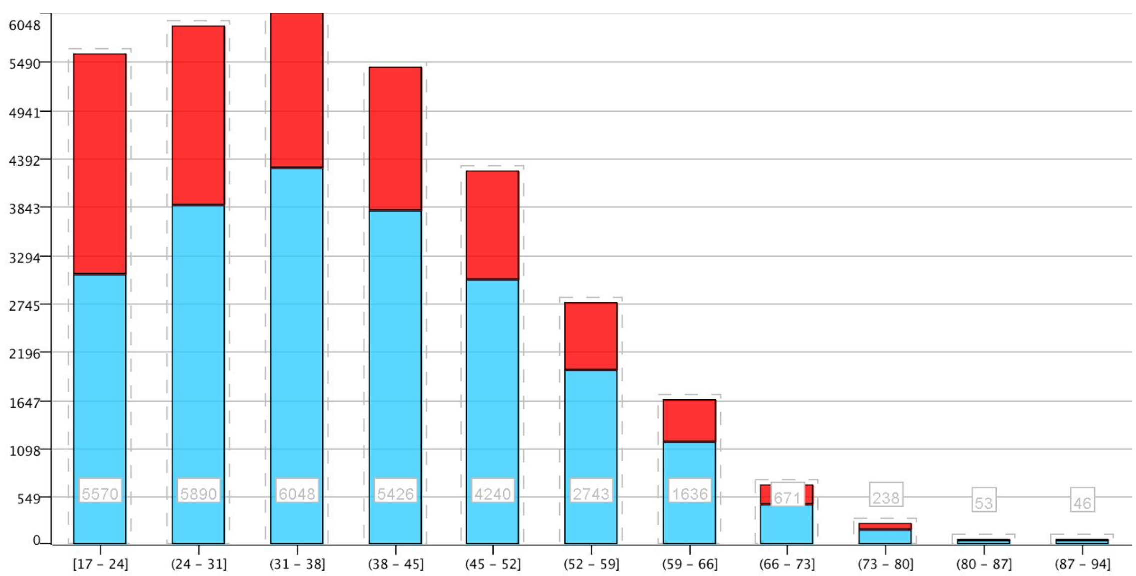
## Summary Statistics

In this section the data is explored visually and statistically to give an insight into the dataset.

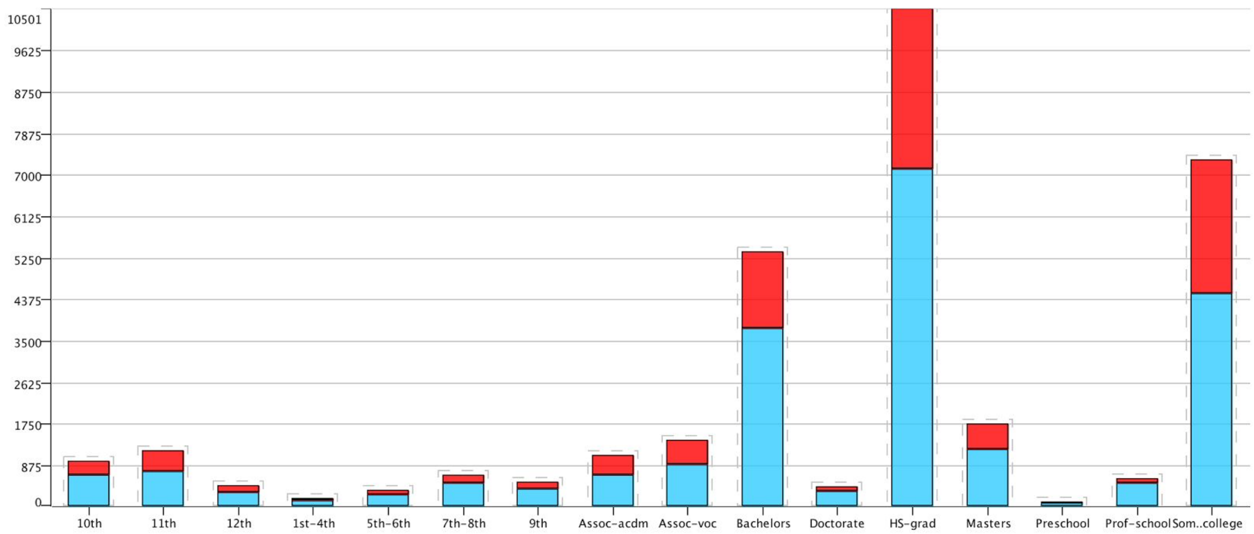
Sex



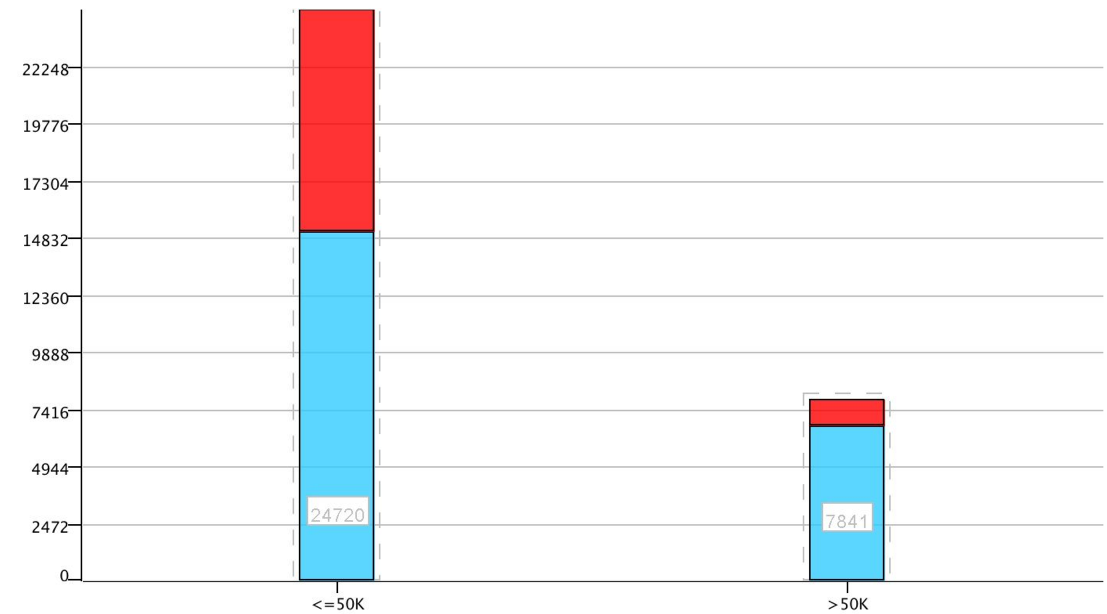

Age



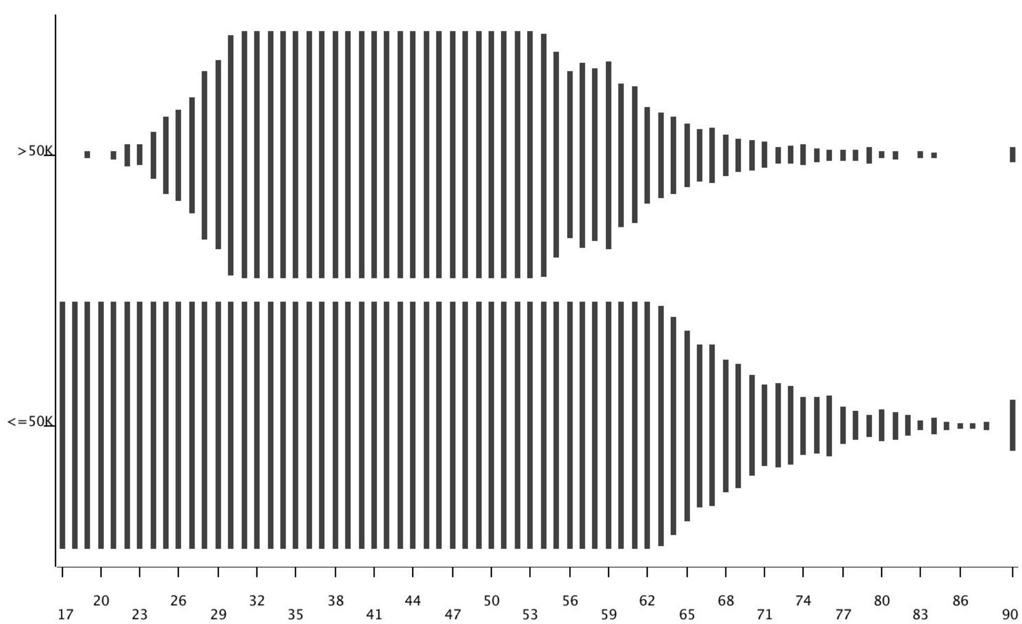
Education


10501
9625
8750
7875
7000
6125
5250
4375
3500
2625
1750
875
0
10th
11th
12th
1st-4th
5th-6th
7th-8th
9th
Assoc-acdm
Assoc-voc
Bachelors
Doctorate
HS-grad
Masters
Preschool
Prof-school
Som..college

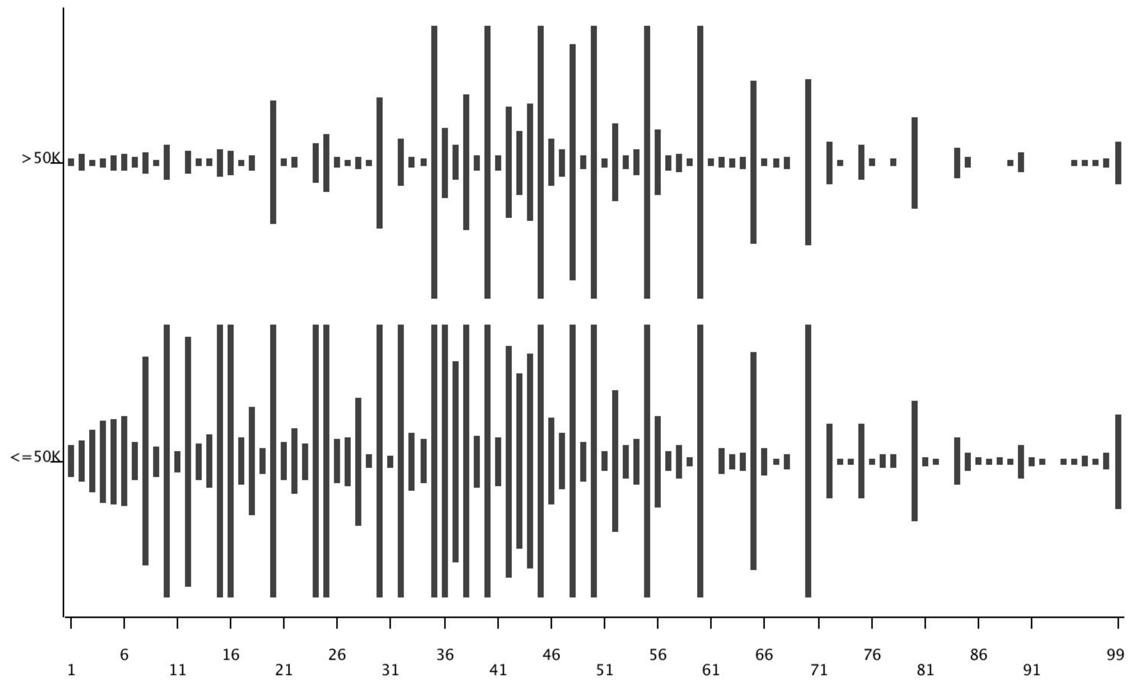
## I.Income



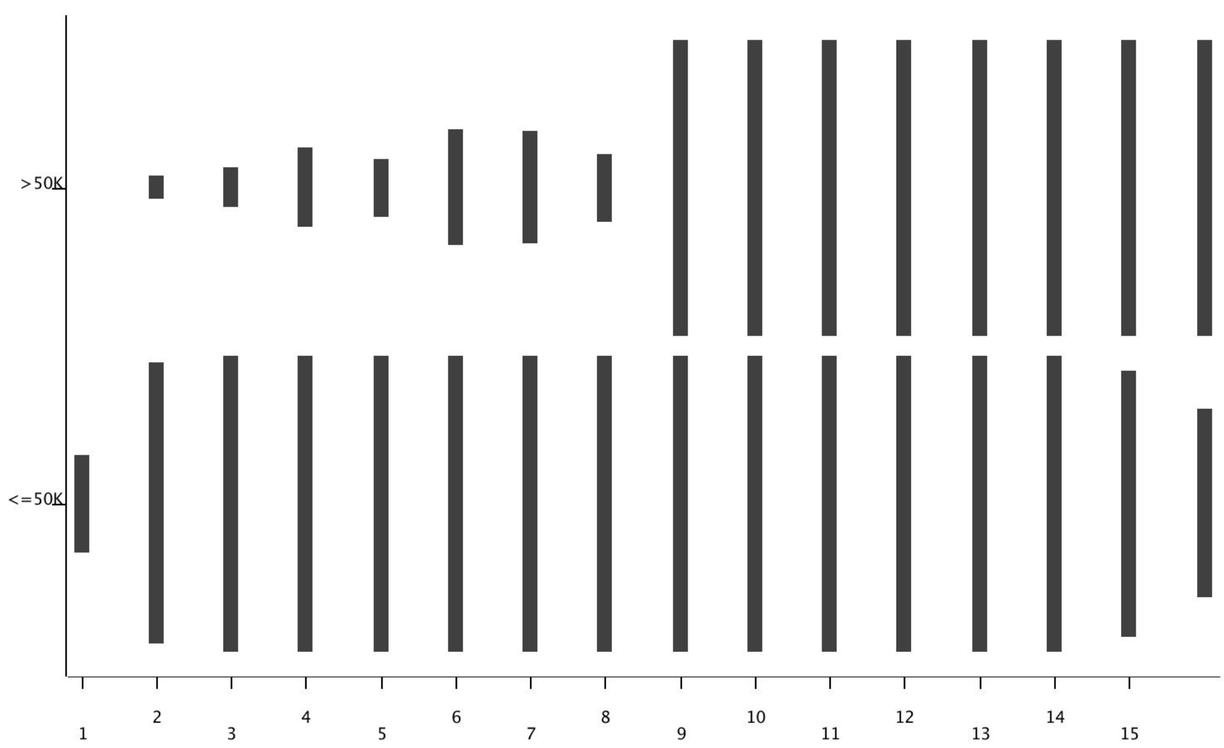

## Age Vs Income



Hours Per Week Vs Income




Education.num Vs Income



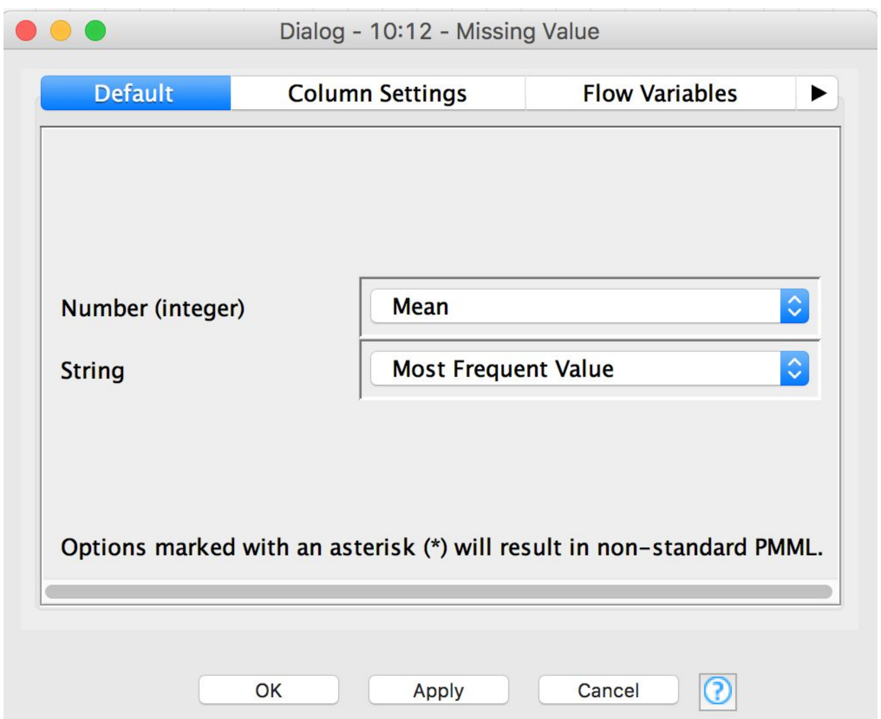
# Approaching the Problem

## Preprocessing

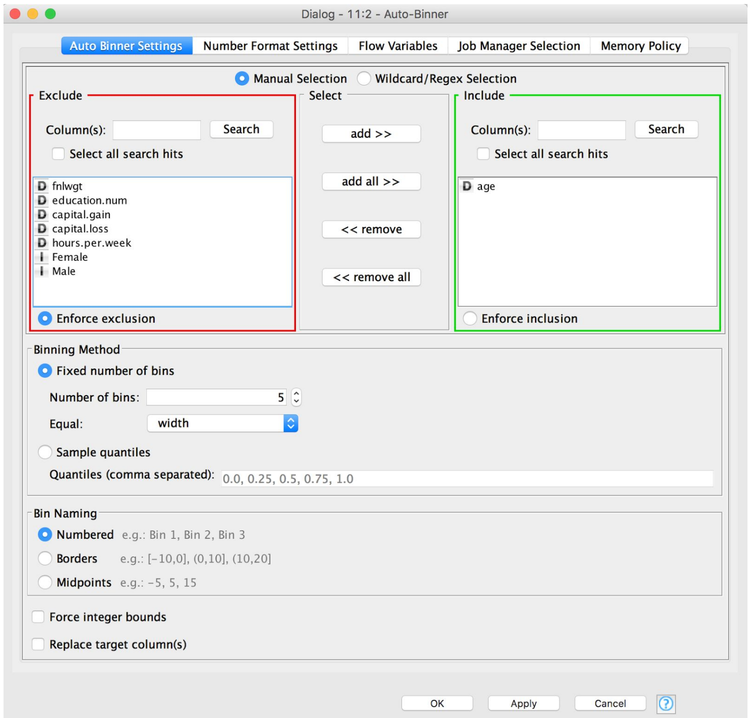
### Eliminate Missing Values

Due to the nature of the initial data collection process, a large number of cases had missing values. The majority of these missing values were located in “workclass” and “occupation”, naturally the occupation is paired with work class therefore if one of them had a missing value the other will also be missing. These missing values made up a very small percentage of the dataset (around 5%), therefore the results should not be harmed due to the deletion.

Further elimination was taken place using the “Missing Value” node. The node was set up to replace the integers with the mean value of the attribute and the strings would be replaced with the most common in that particular attribute. This method was applied to all attributes to effectively clean the data and ensure the best results.

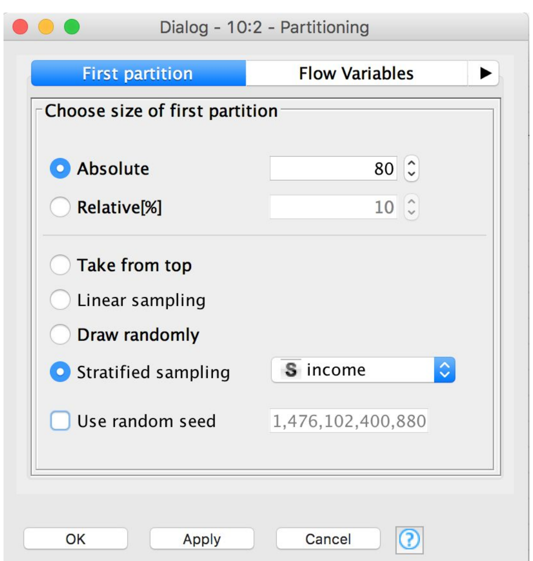


## Binning (age)

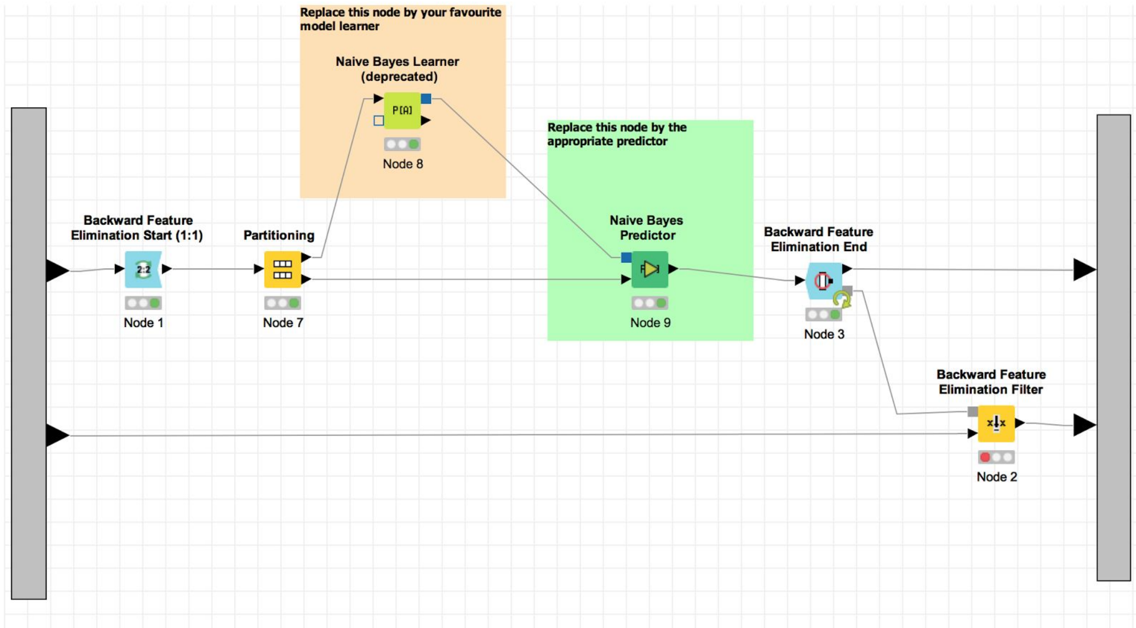



Binning was applied to the age attribute using the Auto - Binner node. Binning allows to reduce noise and account for outliers.

## Partitioning



## Feature Selection




Levels - 10:8:3 - Backward Feature Elimination End

File

Table "default" – Rows: 14 | Spec – Columns: 3 | Properties

| Row ID | Nr. of features | Error ... | Removed feature |
|---|---|---|---|
| All | 14 | 0.209 | |
| 13 | 13 | 0.2 | relationship |
| 12 | 12 | 0.193 | workclass |
| 11 | 11 | 0.19 | race |
| 10 | 10 | 0.187 | native.country |
| 9 | 9 | 0.184 | occupation |
| 8 | 8 | 0.181 | hours.per.week |
| 7 | 7 | 0.181 | fnlwgt |
| 6 | 6 | 0.181 | age |
| 5 | 5 | 0.181 | sex |
| 4 | 4 | 0.178 | education |
| 3 | 3 | 0.18 | capital.loss |
| 2 | 2 | 0.186 | capital.gain |
| 1 | 1 | 0.22 | marital.status |

The backwards feature elimination method was applied to the dataset for feature selection. Feature selection assists in improving prediction performance of a classifier, simplifying a model for easier interpretation as well as reducing run time and CPU usage of a model due to the reduced dimensionality. Backwards feature elimination functions by running a classifier model multiple times to calculate the error rate as a feature is removed from the model. In this case we see the error rate dropped from 0.209 to 0.178 by removing the features relationship, workclass, race, native.country, occupation, hours.per.week, fnlwgt, age and sex.

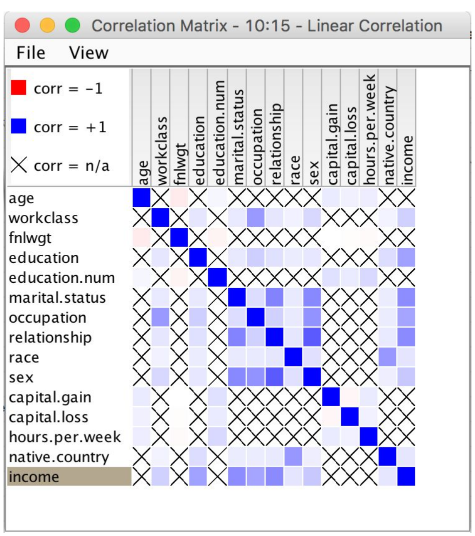
## Numerical Conversion for SVM

| age | workclass | fnlwgt | education.num | marital.status | occupation | relationship | race | sex | capital.gain | capital.loss | hours.per.week | native.country | Male | Income |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 90 | 1 | 77053 | 9 | 1 | 1 | 1 | 1 | 2 | 0 | 1 | 40 | 1 | 0 | 0 |
| 82 | 1 | 132870 | 9 | 1 | 2 | 1 | 1 | 2 | 0 | 1 | 18 | 1 | 0 | 0 |
| 66 | 1 | 186061 | 10 | 1 | 1 | 2 | 2 | 2 | 0 | 1 | 40 | 1 | 0 | 0 |
| 54 | 1 | 140359 | 4 | 2 | 4 | 2 | 1 | 2 | 0 | 0.8953168044 | 40 | 1 | 0 | 0 |
| 41 | 1 | 264663 | 10 | 3 | 1 | 3 | 1 | 2 | 0 | 0.8953168044 | 40 | 1 | 0 | 0 |
| 34 | 1 | 216864 | 9 | 2 | 5 | 2 | 1 | 2 | 0 | 0.8654729109 | 45 | 1 | 0 | 0 |
| 38 | 1 | 150601 | 6 | 3 | 6 | 2 | 1 | 1 | 0 | 0.8654729109 | 40 | 1 | 1 | 0 |
| 74 | 2 | 88638 | 16 | 4 | 1 | 4 | 1 | 2 | 0 | 0.8455004591 | 20 | 1 | 0 | 1 |
| 68 | 3 | 422013 | 9 | 2 | 1 | 1 | 1 | 2 | 0 | 0.8455004591 | 40 | 1 | 0 | 0 |
| 41 | 1 | 70037 | 10 | 4 | 7 | 2 | 1 | 1 | 0 | 0.6896235078 | 60 | 1 | 1 | 1 |
| 45 | 1 | 172274 | 16 | 2 | 1 | 2 | 2 | 2 | 0 | 0.6896235078 | 35 | 1 | 0 | 1 |
| 38 | 4 | 164526 | 15 | 4 | 1 | 1 | 1 | 1 | 0 | 0.6483011938 | 45 | 1 | 1 | 1 |
| 52 | 1 | 129177 | 13 | 1 | 5 | 1 | 1 | 2 | 0 | 0.6483011938 | 20 | 1 | 0 | 1 |
| 32 | 1 | 136204 | 14 | 3 | 2 | 1 | 1 | 1 | 0 | 0.6483011938 | 55 | 1 | 1 | 1 |
| 51 | 1 | 172175 | 16 | 4 | 1 | 1 | 1 | 1 | 0 | 0.6483011938 | 40 | 1 | 1 | 1 |
| 46 | 1 | 45363 | 15 | 2 | 1 | 1 | 1 | 1 | 0 | 0.6483011938 | 40 | 1 | 1 | 1 |
| 45 | 1 | 172822 | 7 | 2 | 8 | 1 | 1 | 1 | 0 | 0.6483011938 | 76 | 1 | 1 | 1 |
| 57 | 1 | 317847 | 14 | 2 | 2 | 1 | 1 | 1 | 0 | 0.6483011938 | 50 | 1 | 1 | 1 |
| 22 | 1 | 119592 | 12 | 4 | 9 | 1 | 2 | 1 | 0 | 0.6483011938 | 40 | 1 | 1 | 1 |
| 34 | 1 | 203034 | 13 | 3 | 10 | 1 | 1 | 1 | 0 | 0.6483011938 | 50 | 1 | 1 | 1 |
| 37 | 1 | 188774 | 13 | 4 | 2 | 1 | 1 | 1 | 0 | 0.6483011938 | 40 | 1 | 1 | 1 |
| 29 | 1 | 77009 | 7 | 3 | 10 | 1 | 1 | 2 | 0 | 0.632231405 | 42 | 1 | 0 | 0 |
| 61 | 1 | 29059 | 9 | 2 | 10 | 2 | 1 | 2 | 0 | 0.632231405 | 25 | 1 | 0 | 0 |
| 51 | 1 | 153870 | 10 | 5 | 8 | 5 | 1 | 1 | 0 | 0.5975665748 | 40 | 1 | 1 | 0 |
| 61 | 1 | 135285 | 9 | 5 | 1 | 5 | 1 | 1 | 0 | 0.5975665748 | 32 | 1 | 1 | 0 |
| 21 | 1 | 34310 | 11 | 5 | 7 | 5 | 1 | 1 | 0 | 0.5975665748 | 40 | 1 | 1 | 0 |
| 33 | 1 | 228696 | 2 | 5 | 7 | 1 | 1 | 1 | 0 | 0.5975665748 | 32 | 20 | 1 | 0 |
| 49 | 1 | 122066 | 3 | 5 | 5 | 5 | 1 | 1 | 0 | 0.5975665748 | 40 | 10 | 1 | 0 |
| 37 | 5 | 107164 | 6 | 4 | 8 | 1 | 1 | 1 | 0 | 0.5874655647 | 50 | 1 | 1 | 1 |
| 38 | 1 | 175360 | 6 | 4 | 1 | 1 | 1 | 1 | 0 | 0.5874655647 | 90 | 1 | 1 | 1 |
| 23 | 1 | 44064 | 10 | 3 | 5 | 1 | 1 | 1 | 0 | 0.5874655647 | 40 | 1 | 1 | 1 |
| 59 | 5 | 107287 | 6 | 1 | 2 | 2 | 1 | 2 | 0 | 0.5874655647 | 50 | 1 | 0 | 1 |
| 52 | 1 | 198863 | 15 | 2 | 2 | 1 | 1 | 1 | 0 | 0.5874655647 | 60 | 1 | 1 | 1 |
| 51 | 1 | 123011 | 13 | 2 | 2 | 1 | 1 | 1 | 0 | 0.5874655647 | 50 | 1 | 1 | 1 |
| 60 | 4 | 205246 | 9 | 4 | 2 | 1 | 2 | 1 | 0 | 0.5874655647 | 50 | 1 | 1 | 1 |

The find and replace function in excel allowed us to replace all strings with a numeric values. This was done specifically for the Support Vectoring Machine Classification to run a successful prediction.

However, income column was converted to a string value using the “Number To String” Node  in order to be able to use it as a target class.

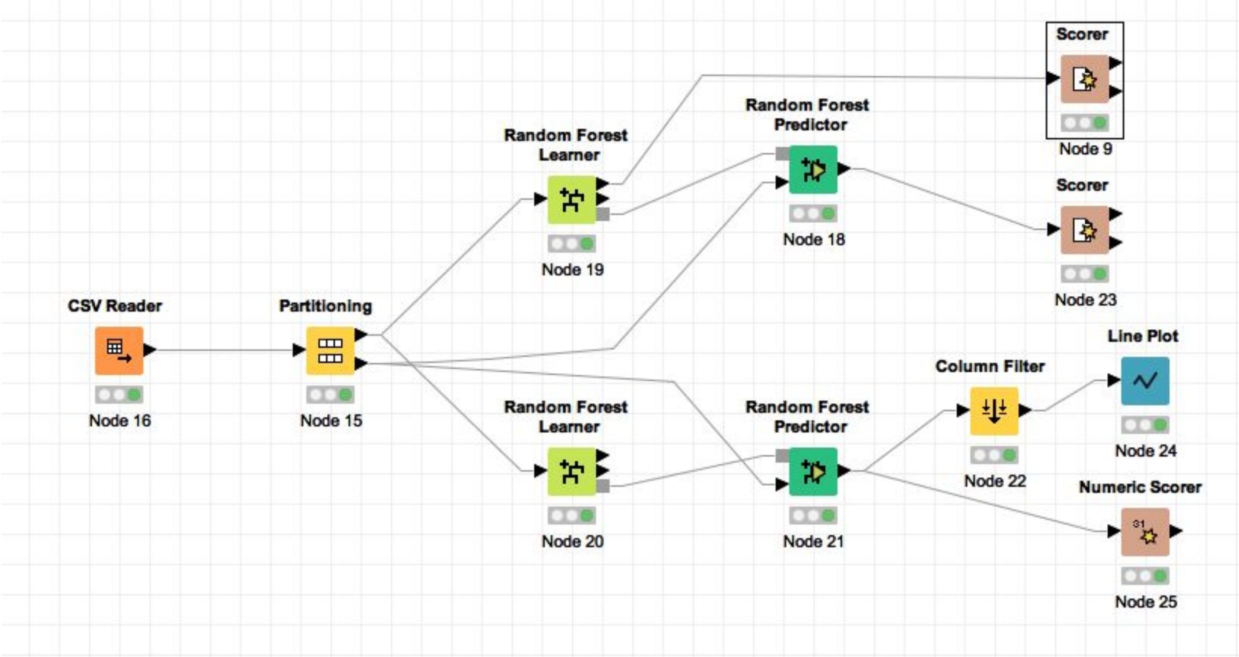
## Linear Correlation



Linear correlation measures how two quantitative variables fluctuate together. Perfect correlation is denoted by 1 and no correlation whatsoever is denoted by -1.  A correlation matrix allows for comparison of all the variables in the dataset. Here color is used to demonstrate the relationship between variables. Variables correlated with the target attribute (income) may be influential attributes for a predictive model.

## Random Forest Classification

### Workflow




### Results

| income \ P... | <=50K | >50K |
|---|---|---|
| <=50K | 18771 | 1005 |
| >50K | 2628 | 3645 |

Correct classified: 22,416　Wrong classified: 3,633

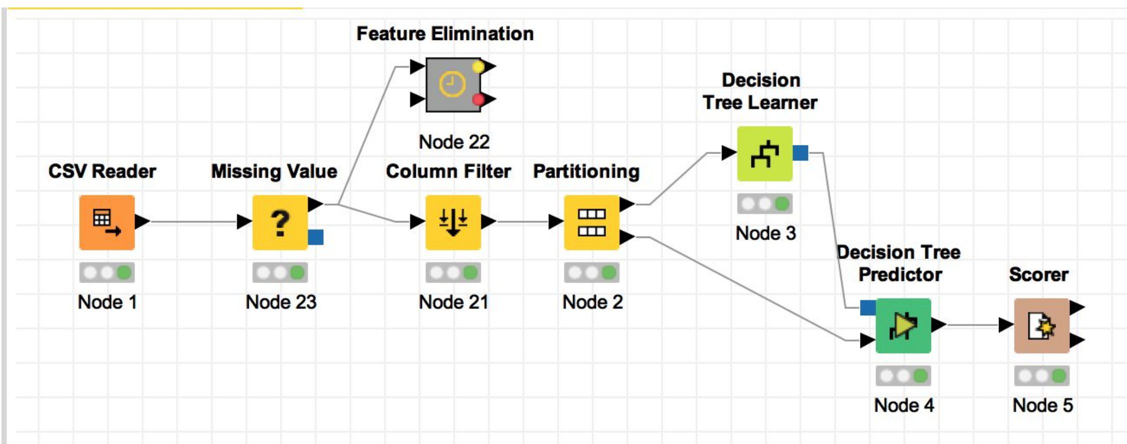
Accuracy: 86.053 %　Error: 13.947 %

Cohen's kappa (κ) 0.582

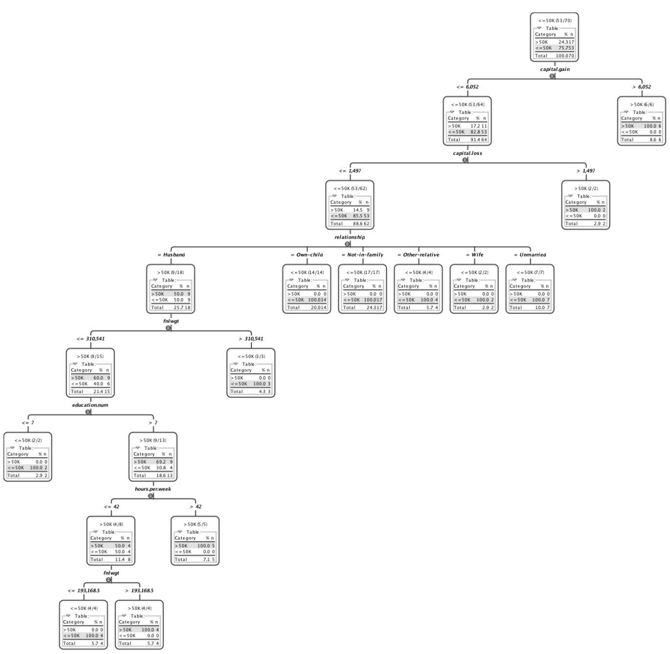
The random forest classifier appears to be the most accurate thus far; it is also the easiest to set up since it works on different types of data. Better results were achieved once the data with unknown values were removed this allowed less room for error within the prediction.

## Decision Tree Classification

### Workflow




### Decision Tree Output



## Results

### No Pre-Processing

Here are the results of feeding the raw data into a decision tree classifier. The classifier performed fairly well achieving an accuracy rate of 78.739%

Confusion Matrix - 9:5 - Scorer

File Hilite

| income \ Prediction (income) | <=50K | >50K |
|---|---|---|
| <=50K | 21036 | 3631 |
| >50K | 3277 | 4547 |

Correct classified: 25,583 Wrong classified: 6,908

Accuracy: 78.739 % Error: 21.261 %

Cohen's kappa (κ) 0.427

### Processed

By feeding the decision tree pre-processed data and applying feature selection the accuracy improved by 1.634%

Confusion Matrix - 9:5 - Scorer

File Hilite

| income \ P... | <=50K | >50K |
|---|---|---|
| <=50K | 23568 | 1102 |
| >50K | 5275 | 2546 |

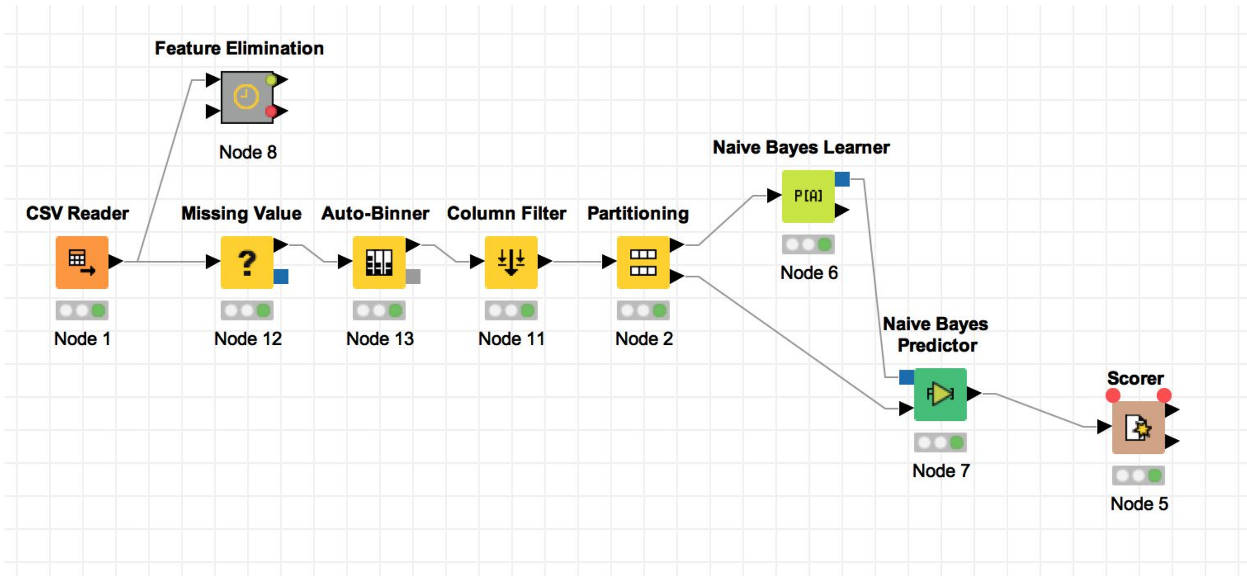
Correct classified: 26,114 Wrong classified: 6,377

Accuracy: 80.373 % Error: 19.627 %

Cohen's kappa (κ) 0.343

# Naive Bayes Classification

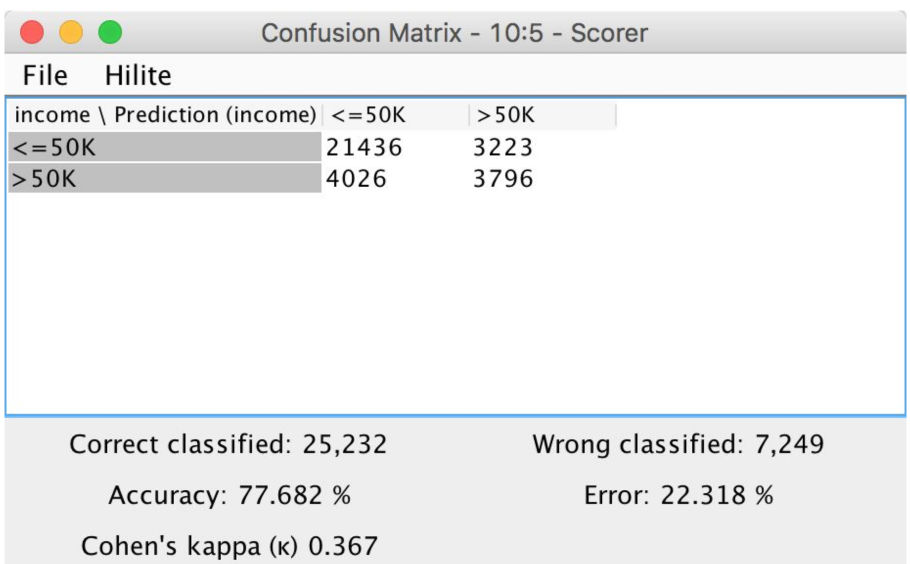
## Workflow




## Results

## No Pre-Processing

A Naive Bayes classifier achieved an accuracy of 77.682% when using raw data.



Confusion Matrix - 10:5 - Scorer

File Hilite

| income \ Prediction (income) | <=50K | >50K |
|---|---|---|
| <=50K | 21436 | 3223 |
| >50K | 4026 | 3796 |

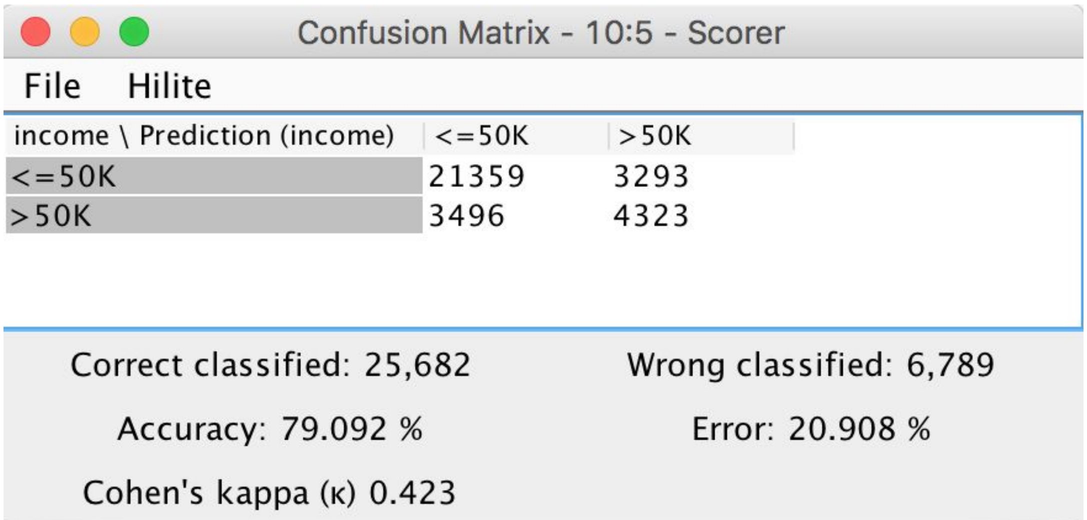
Correct classified: 25,232 Wrong classified: 7,249

Accuracy: 77.682 % Error: 22.318 %

Cohen's kappa (κ) 0.367

### Processed

By applying feature selection to the model and replacing missing values with the mean for quantitative attributes and the mode for categorical attributes the classifier accuracy increased by 1.41%


Confusion Matrix - 10:5 - Scorer

File Hilite

| income \ Prediction (income) | <=50K | >50K |
|---|---|---|
| <=50K | 21359 | 3293 |
| >50K | 3496 | 4323 |

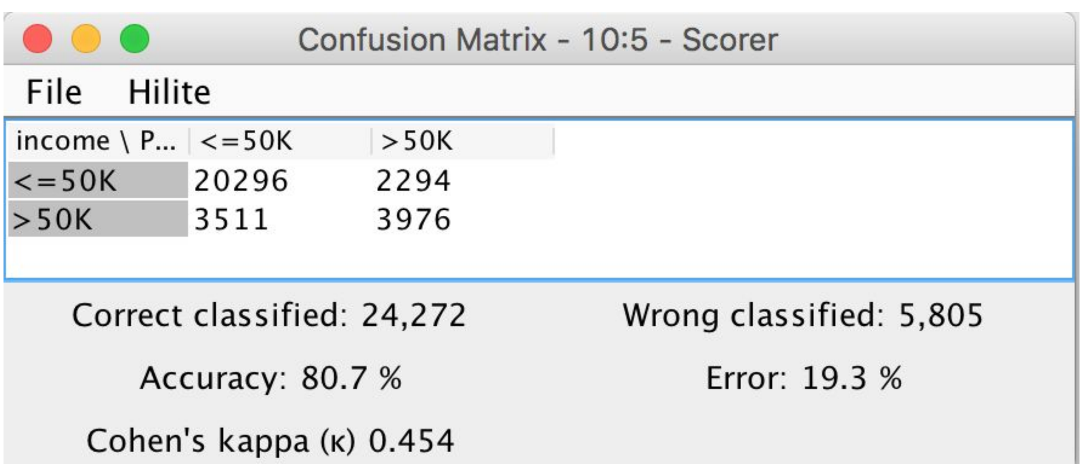
Correct classified: 25,682 Wrong classified: 6,789

Accuracy: 79.092 % Error: 20.908 %

Cohen's kappa (κ) 0.423

### Removing rows of string type with missing values

By altering the missing values handling by removing cases where a string type values was missing classifier accuracy was improved by 1.608%


Confusion Matrix - 10:5 - Scorer

File Hilite

| income \ P... | <=50K | >50K |
|---|---|---|
| <=50K | 20296 | 2294 |
| >50K | 3511 | 3976 |

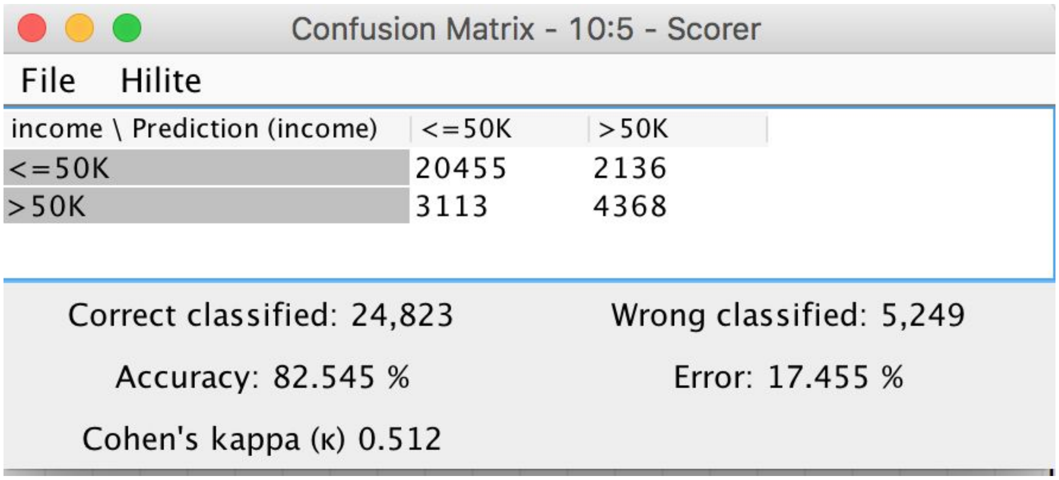
Correct classified: 24,272 Wrong classified: 5,805

Accuracy: 80.7 % Error: 19.3 %

Cohen's kappa (κ) 0.454

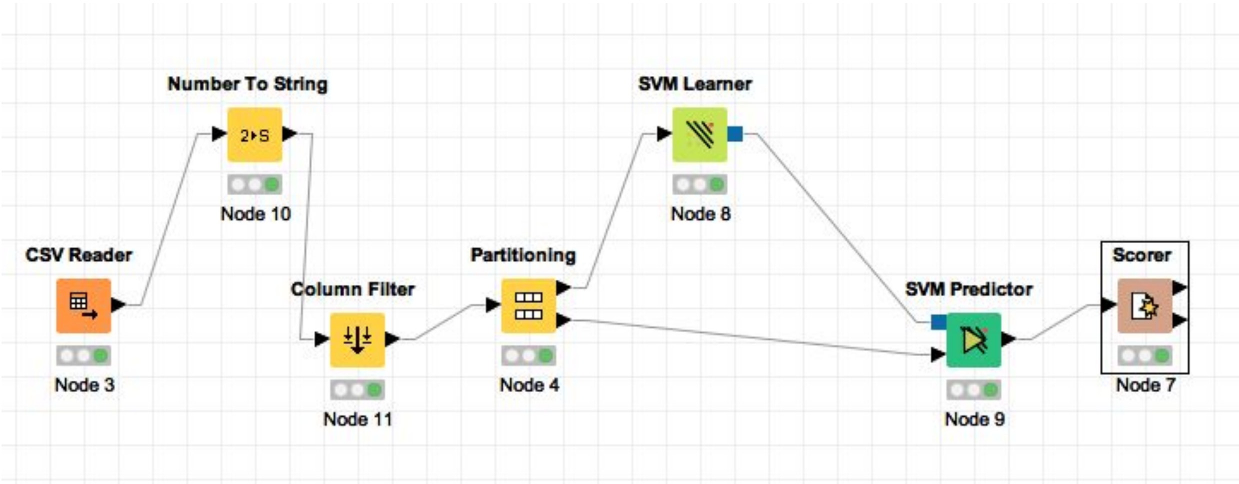
## Removing all rows with missing values and partitioning by random drawing

A further 1.845% accuracy was achieved by altering the partitioning settings and missing value handling. The partitioning method was changed to random drawing from stratified sampling of income and all cases with a missing value were removed from the model.



# Support Vector Machine Classification

## Workflow




## Results

| Income \ P... | 0 | 1 |
|---|---|---|
| 0 | 19191 | 599 |
| 1 | 4097 | 2162 |

Correct classified: 21,353 Wrong classified: 4,696

Accuracy: 81.972 % Error: 18.028 %

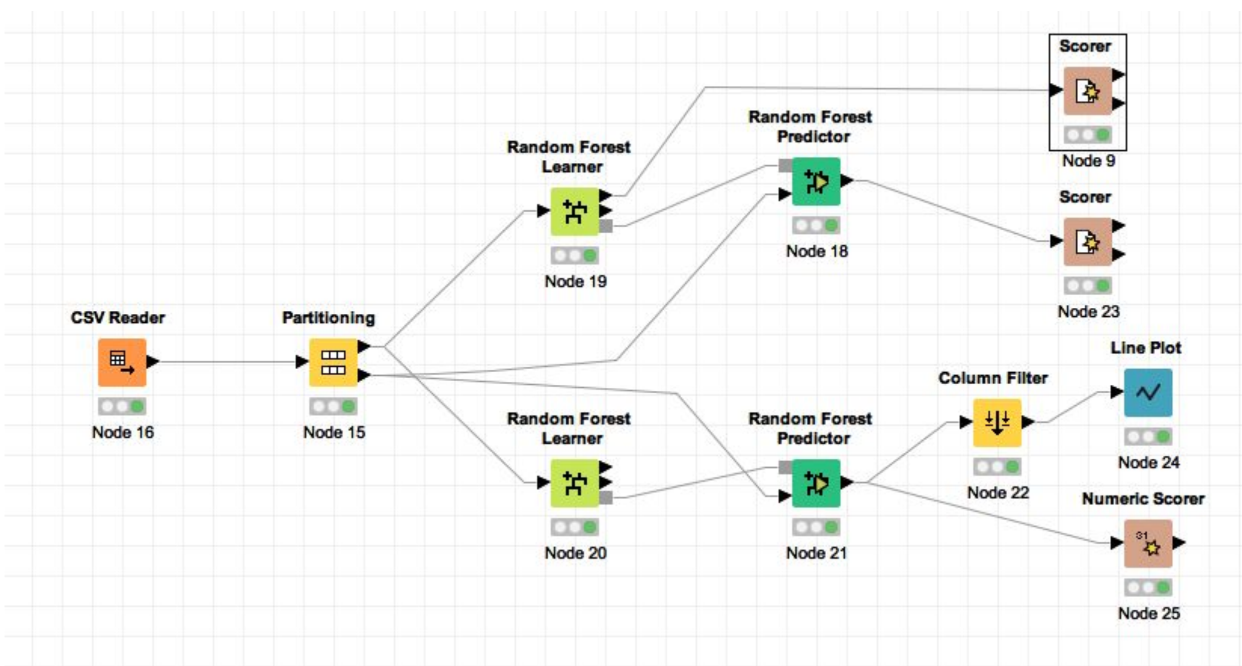
Cohen's kappa (κ) 0.39

The support vector machine required a great deal of data preprocessing. All of the strings needed to be changed to integers for the classifier to effectively predict the data. All recurring string cells were converted to the same number to ensure accuracy in the prediction. The class column was then converted to a string to enable to SVM to use it as a class predictor.

# Recommended Classifier

After the dataset was processed we applied the data to a number of classifiers using Knime. The majority of the results were shown to be very similar; however, the random forest classifier proved to be more accurate. With an accuracy percentage of 86.053% it was significantly better than the other methods. A close second method was Naive Bayes with an accuracy rating of 82.545%, a very acceptable prediction.

Random Forest also showed to be the most versatile in terms of analysing different types of data, it proved to work even without preprocessing although the percentage was significantly lower. It was able to run with a mixture of string and numeric values and still provide a positive result.




This method shows great potential, with very little preprocessing it was still able to provide a high accuracy rating. In this particular case we applied a number of preprocessing techniques to provide better results. These include; Missing value elimination, Binning, Partitioning and Feature Selection. Many more tools can be used to process the data and reach a higher level of accuracy.