DATA MINING

Mining A Dataset

By Jeffrey Farnan

Table of Contents

Business Understanding

This dataset was extracted from the 1994 census bureau database. It's a collection of data that comprises of information about people between the ages of 16 to 100. The dataset contains 32561 rows of data and 14 columns containing information about people's age, sex, income, education, marital status and where they were born, etc. The Data mining objective is to mine the dataset using the Crisp-Dm framework. The Business objective is to use information gained to help with project planning and future business needs

Accomplish

- To mine the dataset and identify useful information patterns that can be used to accurately predict who will earn over 50k based on the data provided.

Objectives

- To better understand and find useful patterns in the datasheet.
- To help read the data with the use of classification.
- Predict who will earn over 50k with high accuracy.

Data Understanding

The first phase of the project is to look at the data set and understand the information being supplied by the data, identify and discover problems or insights the data might provide, so an assessment can be made on the quality of the data.

Doing an initial survey

From an initial glance of the Meta Data of the dataset statistics, I can describe the information content as good quality, with 32561 examples (rows) and with 15 columns, containing 14 regular attributes and 1 special attribute. When looking at the dataset more closely I seen that many missing values were not recognized by Rapid Miner, it only recognized 4 but there were many more with a (?) which were not detect as missing. I deleted these in excel and imported in the dataset again.

Describing the data

To describe the data we will look at its Meta Data . A table view of each attribute, its description and data type.

ROLE	NAME	TYPE	STATISTICS	RANGE	MISSING
label: This is the attribute that the algorithms try to predict	**label: or (income)** this attribute indicates whether they earn 50k and above or earn below 50K	**binominal:** This attribute has 2 possible values. <=50K below or equal to 50k. >50K everything above 50k	**mode = <=50K (24720), least = >50K (7841)**	**<=50K (24720), >50K (7841)** This attribute shows the number of occurrence of each value.	**0** number of missing attributes
regular: used to predict the class label	**age:** the age of the person	**real:** holds a floating point number	**avg = 38.582** The average value **+/- 13.640** The standard deviation	**[17.000 ; 90.000]**	**0** number of missing attributes

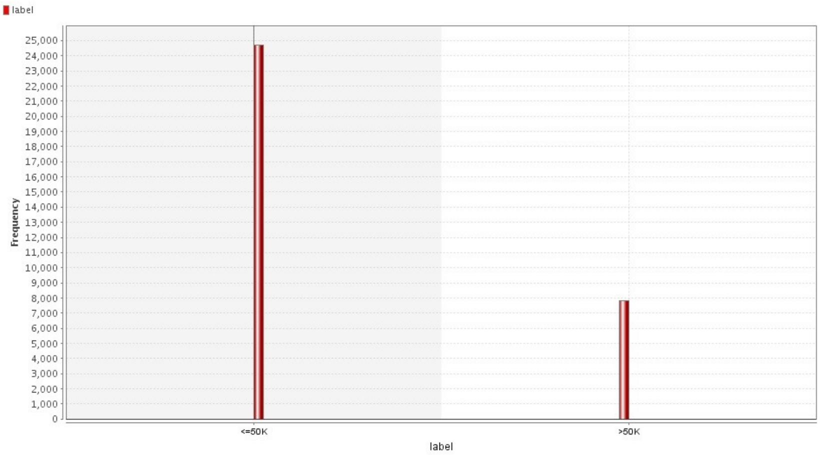
regular: used to predict the class label	**workclass:**	**polynominal:** several number of variable length alphanumeric values	**mode = Private (22696), least = Never_worked (7)**	**State_gov (1298), Self_emp_not_inc (2541), Private (22696),Federal_gov (960), Local_gov (2093), ? (1836), Self_emp_inc (1116), Without_pay (14), Never_worked (7)** shows the number of occurrence of each value.	**1836** number of missing attributes
regular: used to predict the class label	**fnlwgt:** (final weight)	**real:** holds a floating point number	**avg = 189778.367** The average value **+/- 105549.978** The standard deviation	**[12285.000 ; 1484705.000]**	**0** number of missing attributes
regular: used to predict the class label	**education:** The highest form of education they got	**polynominal:** several number of variable length alphanumeric values	**mode = HS_grad (10501), least = Preschool (51)**	**Bachelors (5355), HS_grad (10501), 11th (1175), Masters (1723), 9th (514), Some_college (7291), 1st_4th (168), Preschool (51), 12th (433)**	**0** number of missing attributes
regular: used to predict the class label	**education_num:** The further up you go in education the higher the number you are given.	**real:** holds a floating point number	**avg = 10.081** The average value **+/- 2.573** The standard deviation	**[1.000 ; 16.000]**	**0** number of missing attributes
regular: used to predict the class label	**marital_status:**	**polynominal:** several number of variable length alphanumeric values	**mode = Married_civ_spouse (14976), least = Married_AF_spouse (23)**	**Never_married(10683),Married_civ_spouse (14976), Divorced (4443),Married_spouse_absent (418), Separated (1025), Married_AF_spouse (23), Widowed (993)**	**0** number of missing attributes
regular: used to predict the class label	**occupation:** The job they work at	**polynominal:** several number of variable length alphanumeric values	**mode = Prof_specialty (4140), least = Armed_Forces (9)**	**Adm_clerical (3770), Exec_managerial (4066),Handlers_cleaners (1370), Prof_specialty Armed_Forces (9),**	**1843** number of missing attributes

				Priv_house_serv (149)	
regular: used to predict the class label	**Relationship:** What kind of relationship are people in.	**polynominal:** several number of variable length alphanumeric values	**mode = Husband (13193), least = Other_relative (981)**	**Not_in_family (8305), Husband (13193), Wife (1568), Own_child (5068), Unmarried (3446), Other_relative (981)**	**0** number of missing attributes
regular: used to predict the class label	**race:** white, black, Asian_Pac_Islander,Amer_Indian_Eskimo, other	**polynominal:** several number of variable length alphanumeric values	**mode = White (27815), least = Other (271)**	**White (27815), Black (3124), Asian_Pac_Islander (1039),Amer_Indian_Eskimo (311), Other (271)**	**1** number of missing attributes
regular: used to predict the class label	**sex:** the sex of the person, male or female	**binominal:** This attribute has 2 possible values.	**mode = Male (21788), least = Female (10770)**	**Male (21788), Female (10770)** This attribute shows the number of occurrence of each value.	**3** number of missing attributes
regular: used to predict the class label	**capital_gain:**	**real:** holds a floating point number	**avg = 1077.649** The average value **+/- 7385.292** The standard deviation	**[0.000 ; 99999.000]**	**0** number of missing attributes
regular: used to predict the class label	**capital_loss:**	**real:** holds a floating point number	**avg = 87.304** The average value **+/- 402.960** The standard deviation	**[0.000 ; 4356.000]**	**0** number of missing attributes
regular: used to predict the class label	**hours_per_week:** hours worked per week	**real:** holds a floating point number	**avg = 40.437** The average value **+/- 12.347** The standard deviation	**[1.000 ; 99.000]**	**0** number of missing attributes
regular: used to predict the class label	**native_country:** country person was born in	**polynominal:** several number of variable length alphanumeric values	**mode =United_States (29170), least = Holand_Netherlands (1)**	**United_States (29170), Cuba(95), Jamaica (81), India(100), ? (583), ..., Netherlands(1)**	**583** number of missing attributes

Explore the data

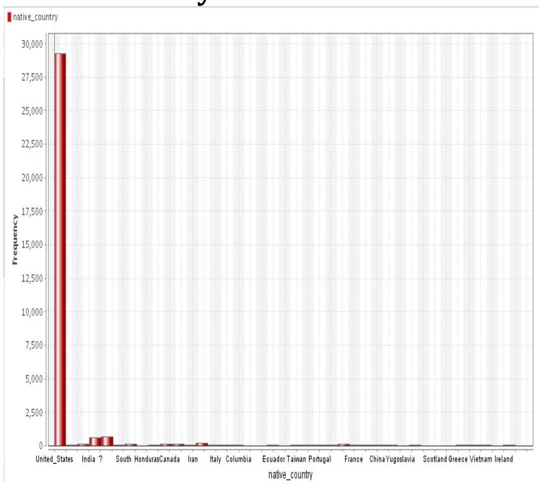
This dataset is vast in its information content with having over 30000 rows of data and 14 columns. By exploring the data, by looking at its information content and using plot view, which allows you to view the data in different visualised formats , I made the following observations.

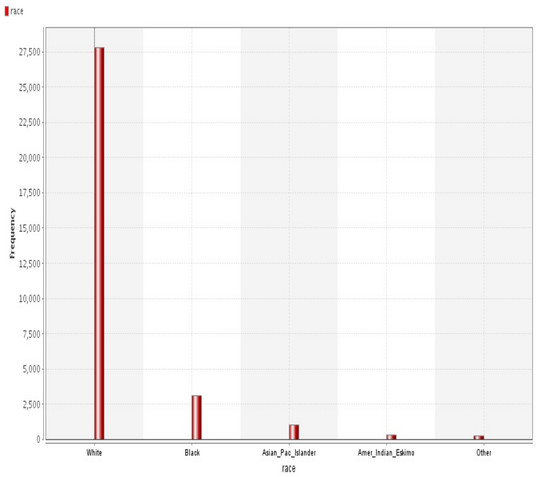
Class Label



This histogram shows that this graph is unbalanced and that the majority of people earn below or equal to 50k, while very little earn above 50k.

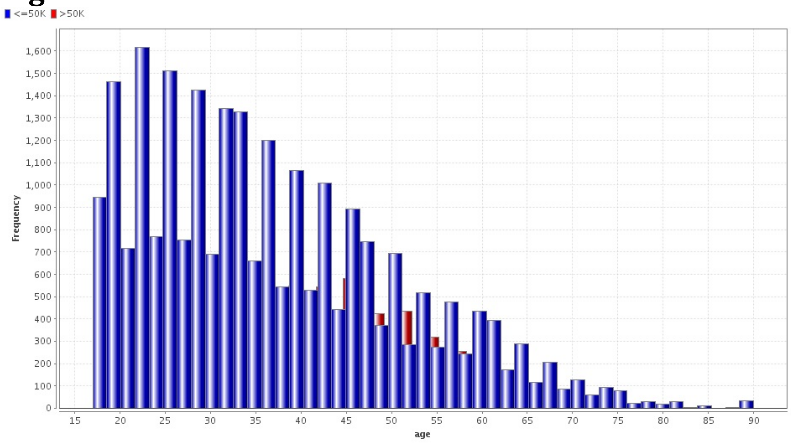
Native Country and Race





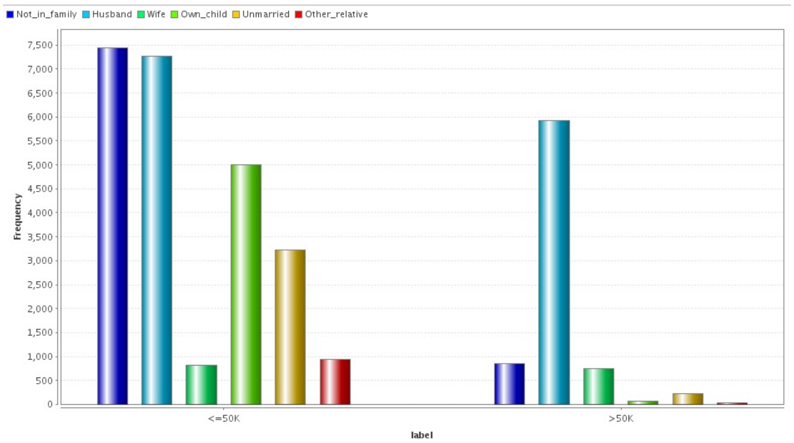
We can see in these histograms are skewed to the left, showing the majority of people in the dataset is white and born in the United States. This dataset is very unbalanced and the predictive outcome will favour white people and those born in the United States.

Income and Age



We can see from this colour histogram the blue columns in the graph that the age range is between 16 and 90 and most earn below 50k, with people earning the most in this range being around 20 to 25. This graph is skewing to the left and unevenly distributed which shows as people get older they earn less, there is a few exceptions, red columns is showing those earning over 50k are between the ages of 40 to 55.

Relationship and Income



In this graph we can clearly see that most people who earn less than 50k are not in a family, and the next is the husband, the least is the wife with the lowest level. Those who earn above 50k are the husbands who have the highest frequency

Capital Gain, Capital loss

There are some variables that don't seem to show a lot of useful information, like the Capital_Gain, Capital_Loss attributes. As the majority of values for these Attributes are zero, these will be of little use.

Education and Education number

These two attributes holds the same information content, the higher the educated level, the higher education number you are given.

Verify data quality

The original dataset had only four missing values one for race and three for sex, that Rapid Miner recognized as missing, these were values that were left blank, but there were many other missing values, which had a question mark (?) filling these values, which rapid miner did not recognize as missing, these had to be taken out. I tried to delete these values in Rapid Miner but found I could not, because Rapid Miner uses the question mark (?) as a special character, by using operators such as Remove or Replace did not help either

I opened the dataset in Excel, using the Find & Select button which allows you to format text, and using the replace tool, I found all the (?) question marks in the file and replaced them with empty values. I saved this file as an excel file and imported it into rapid miner again. Now rapid miner can spot all the missing values.

The data set has a total of 4266 missing values. The Race Column is missing 1 value, the Sex Column is missing 3, the Native Country is missing 583 and the Occupation Column is missing 1843 values and workclass is missing 1836 values.

There is no presence of noise, bias or outliers which would be likely to cause an issue.

Data Preparation

Before starting the pre-processing the data, I needed to get a baseline accuracy using an x-validation block with a decision tree. The baseline accuracy was 83.95%, recall and precision details are also shown in the confusion matrix below.

accuracy: 83.95% +/- 0.48% (mikro: 83.95%)

	true <=50K	true >50K	class precision
pred. <=50K	23288	3795	85.99%
pred. >50K	1432	4046	73.86%
class recall	94.21%	51.60%	

Select Data

With having such a vast dataset I needed to reduce the rows somehow while still gaining a dataset that still contained the original patterns, I could do this by talking a sample of about 1000 rows and using this sample to run the algorithms on, but first I needed to deal with the missing values. I could take a sample from the dataset but this would likely to contain missing values which I would have to replace somehow, this seemed pointless to me when there was already enough good rows to choose from, so I decided to remove the missing values, useless columns and duplicate values than take a sample.

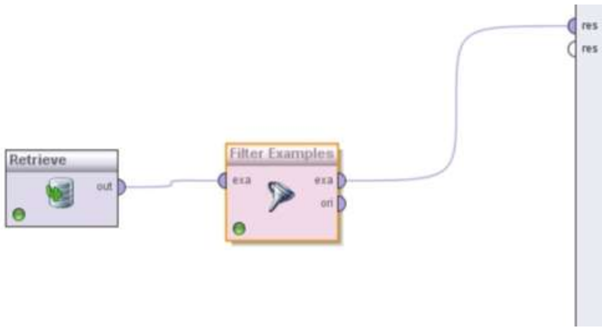
Removing missing values

There are a number of missing values in the data set:

- **Race**(1 missing value)
- **Sex** (3missing values)
- **Native Country** (583 missing values)
- **Workclass** (1836 missing values)
- **Occupation** (1843 missing values)

By removing these rows should not affect the overall outcome.

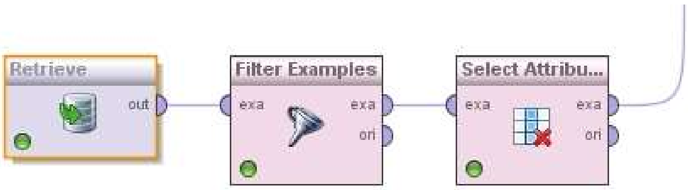
To remove rows I selected the Filter Examples operator, and dragged it onto my process. I selected missing_attributes from the drop down box and clicked on the Invert Filter Box. After running the process, I was left with 30158 rows left.



I checked the accuracy again which was is 83.52% by removing the missing rows it did not have a huge impact on the overall result. In fact very little had changed.

Remove useless columns

By using the Select Attribute you can ignore the Columns you don't want to use. I'm going to use this operator on Capital Gain and Capital loss because of the amount of useless information in these columns. I added the Select Attributes to my process and ran it.



After deleting columns the accuracy is still 83.52 and no changes occurred. This attribute didn't affect the accuracy at all. It remained the same as it was before.

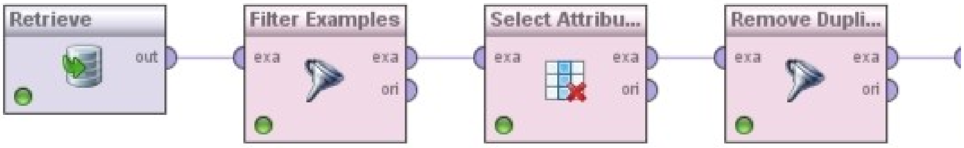
Remove Duplicate Values

To remove duplicate values in the dataset I placed the Remove duplicates operator onto my process window. Rapid Miner removed 27 duplicate values in the dataset which brings the dataset rows down to 30131.

After removing these values I checked for accuracy again. The new baseline Accuracy is now 81.12%

Sampling the dataset

At this point the dataset has still over 30000 rows, it does not need this many rows to be accurate. I placed a sample operator in the process and set the sample size to 1000 (in absolute mode) .



After taking a sample I check to see if the same patterns in the data as before by viewing the except same graphs in plot view, as before when exploring the dataset, giving the same results but less data. Checked for accuracy again and got 80.60%

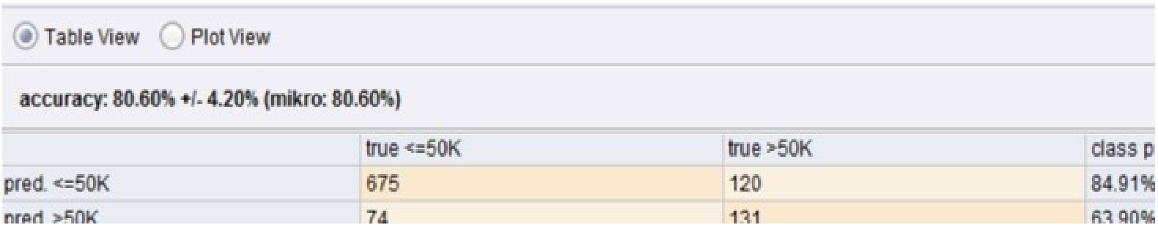


Table View Plot View

accuracy: 80.60% +/- 4.20% (mikro: 80.60%)

	true <=50K	true >50K	class p
pred. <=50K	675	120	84.91%
pred. >50K	74	131	63.90%

Clean Data

I wanted to remove attributes that I felt was not relevant or found not to be useful in making a predication. Which were?

- **Age:** I did not this holds any value to the final outcome
- **Education:** This attribute holds the same information as Education Number
- **Hours per week:** I did not this holds any value to the final outcome

Added Select Attributes to the process and selected **age**, **hours per week** and **education** as the attributes to remove. After running an x-validation block I got an accuracy of 81.30

accuracy: 81.30% +/- 3.61% (mikro: 81.30%)

	true <=50K	true >50K
pred. <=50K	694	132
pred. >50K	55	119

From the drop down accuracy of 80.60% after taking a sample of 1000 rows, the accuracy has now improved to 81.30%.

Removing attributes which might improve accuracy

I wanted to try and improve the accuracy by removing some attributes which I thought might not be working. I changed *education* to *education_number* to see if any improvement. The accuracy dropped down to 77.10% so I brought back *education_number.* I wanted to see if removing *final weight* would have any effect, the accuracy dropped down from 81.30 to 81.20. I removed *marital_status* and the accuracy improves to 80.80.

Table View Plot View

accuracy: 80.80% +/- 3.71% (mikro: 80.80%)

	true <=50K	true >50K
pred. <=50K	696	139
pred. >50K	53	112

When *sex* was removed the accuracy went to 80.90. Although when sex was removed the accuracy improved, I decided to put it back in because I thought it would be important to the final outcome.

Modelling

For the modelling section I will present the different modelling techniques I tried and attempt to interpret the results of these models

Select modelling technique

The classification algorithms 'learn' the patterns of a dataset in different ways, so I chosen the algorithms those suitable for this dataset's attributes, which are Naive Bayes and K-Nearest Neighbour. Naive bayes are good for small data sample, easy to create and need less training data. The K-Nearest Neighbour Algorithm allows you to can change the values to try different K values

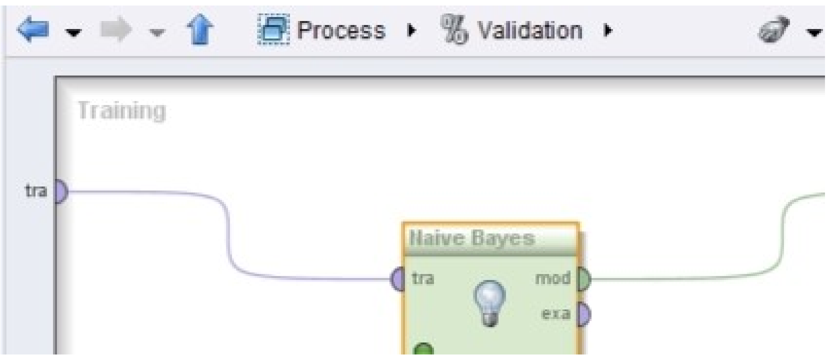
Generate Test Design

To generate training and test data I used cross-validation because it uses the whole dataset for both training and testing.

Build and assess the model

Naive Bayes

Naive Bayes classifier calculates the probability of a row being in a particular class based on its attribute values. This algorithm can look for one feature and if that feature is parent or not it can then classify that data. This algorithm works with all attributes, but prefers categorical attributers and a categorical label.



The Naive Bayes algorithm gave us an accuracy of 76.67%, this accuracy is good but not as good as the accuracy I got for the decision tree. The confusion matrix is shown in the figure below.

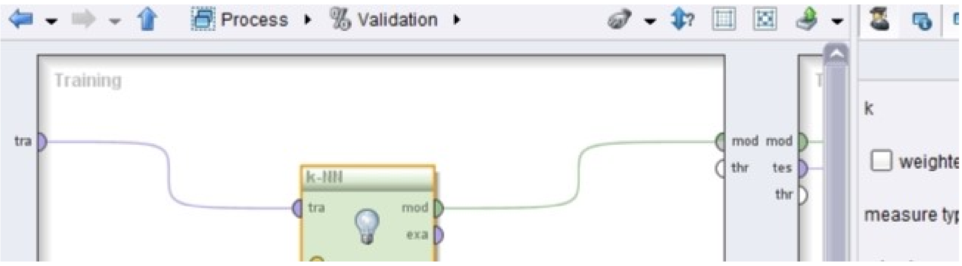
Table View Plot View

accuracy: 76.67% +/- 4.41% (mikro: 76.64%)

	true <=50K	true >50K
pred. <=50K	579	69
pred. >50K	163	182

K-Nearest Neighbour

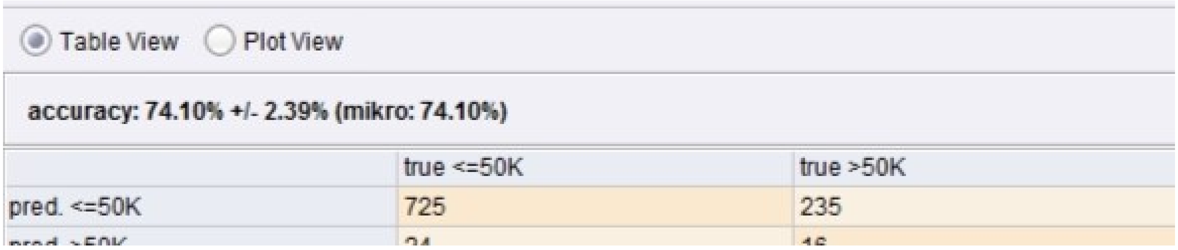
KNN is a simple but fundamental method of classification, it's versatile and can be used in many situations. KNN is a lazy learner algorithm; it does not use training and does not make assumptions on the distribution of the underlying data. It gives the user to change to alter the search parameters allowing different parameters to be used. The value of K can be changed allowing you to compare more neighbours together and increase or decrease the accuracy according to the value set by k. After setting the KNN test with Rapid Miner, I ran the test several times, changing the value for k each time.



The accuracy's I got by changing the k values are: **k1** = 64.10%, **k2** =73.20%, **k3**= 70.20%, **k4**= 73.40%, **k5**=70.80%, **k6**= 74.10%, **k7**= 72.00%, **k8**= 73.80%, **k9**= 72.20%.

The best accuracy was gotten using k set to 6, which gave us an accuracy of 74.10%. k1 got the lowest accuracy with 61.10%, possibly because it was interpreting noise, which would give a low accuracy. Accuracy increased as I moved up the k values, but fluctuating between 70 and 73 and then after k6 dropping down again fluctuating between 70 and 73.

The confusion matrix for k6 is shown in the figure below.



	true <=50K	true >50K
pred. <=50K	725	235

Evaluation

The purpose of this project was to mine the dataset containing information that was contained in the 1994 census bureau database, and use this information to predict who earns over 50k.

The dataset was vast and quite extensive in its information content. From exploring the dataset I found missing values, duplicate Values, and attributes that were not much use in information content, so I used filtering techniques to remove them from the dataset and then I took a sample of 1000 rows, and used that sample as my dataset.

From exploring the dataset I found that several attributes did not help with my objective which was Age, Education, Hours per week and Marital Status,, by filtering out these attributes allowed me to concentrate my efforts on the data that I found to be the most relevant and useful to be mined.

The two algorithms predicted the outcome with slightly different accuracy, with k-NN getting accuracy of 74.10%. And Naive Bayes getting the best accuracy of 76.67%. The Native Bayes classifier allows you to view plots and graphs of each attribute which can be analyzed and interpreted, allowing me to come to following conclusions about who earns over 50K

- The majority are white men aged between 40 to 55,
- Work in the private sector in Executive or Managerial positions
- Have a Bachelors Degree work in the private sector,
- Are married or in civil relationship
- Live in and were born in the United States.

I believe that this project has proven that it is possible to help predict who earns over 50K and gained valuable information that will help with project planning and future business needs.