Data Mining Project

Team

Daniel Hanspeter 6129

Daniel Graziotin 4801

Thomas Schievenin 5701

Datasets:

**Blood transfusion**

**Census-Income**

1. Blood transfusion
2. Census-Income

# The description of the datasets

1. Blood transfusion

This data is as the name says a collection of data taken from blood transfusions and to be more precise, it was collected at the Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The data is collected randomly from donors and is used to achieve a Frequency-Metric-Model of donations, focusing on March 2007.

The data collected focuses on the donor’s situation, namely on the amount of done donations before, of the total amount of donations and so on, for a total of 5 Attributes. The data collected is focused on March 2007, since one attribute collects exactly if one person donated in March 2007 or not. All collected data is numerical, except for the donation in March ’07 which is a bit / Boolean attribute.

No missing values in the Dataset.

1. Census-Income

This data is a census done by the US Census Bureau investigating about the income class of randomly taken people, taking in account their education, current living situation and so on.

The data is described by 14 attributes detailing each person’s situation (i.e. martial-status). The attributes type vary, since some are numerical values as the age, and others are explicit defined fields (nominal), such as the native country (i.e. United-States)

It appears that two attribute listed are mirrors of one another. Education and Education-Num, where Education-Num is a numeric representation of the other.

There is an attribute called *fnlwgt* which seems to have no relation to the income. Therefore it has no predictive power and can be ignored.

Some attribute appear to have an imbalanced distribution of values, as for instance *Age, education, capital-gain* and *capital-loss* which arevery skewed towards lower values

Around 7% of Attributes are missing. To work with that data, we can easily set a default value, like for instance an average if the value is continuous or the most occurring in order to not affect statistics.

# The preprocessing operations performed on the data: missing values, outlier detections (possibly removal)

1. Blood transfusion

|  |  |  |
| --- | --- | --- |
| Label | Frequency (times) | Monetary (c.c. blood)  As derivable from the Boxplots in both collected values there are Positive outliers.  For the Monetary (c.c. blood) Boxplot we can say that the values are located in a very short range, by noticing that the whiskers are just below/above the box ranges.  No missing values in the Dataset. |
| Min | 1 | 250 |
| Q1 | 2 | 500 |
| Median | 4 | 1000 |
| Q3 | 7 | 1750 |
| Max | 50 | 12500 |
| IQR | 5 | 1250 |
| Upper Outliers | 45 | 45 |
| Lower Outliers | 0 | 0 |
| *For the Box (IQR and Median)* | |  |
| Q2-Q1 | 2 | 500 |
| Q3-Q2 | 3 | 750 |
| *For the Whiskers* | |  |
| Q3+1.5\*IQR | 14,5 | 3625 |
| Q1-1.5\*IQR | -5,5 | -1375 |
| Upper Whisker | 14,5 | 3625 |
| Lower Whisker | 1 | 250 |
| Wupper-Q3 | 7,5 | 1875 |
| Q1-Wlower | 1 | 250 |
| *For the Outliers* | |  |
| Max | 50 | 12500 |
| Min | #N/A | #N/A |

1. Census-Income

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Labels | Education-num | hours-per-week |  |  |  |  |  |  |  |  |  |  |  |
| Min | 1 | 1 |  | | | | | | | | | | |
| Q1 | 9 | 40 |
| Median | 10 | 40 |
| Q3 | 12 | 45 |
| Max | 16 | 99 |
| IQR | 3 | 5 |
| Upper Outliers | 0 | 3492 |
| Lower Outliers | 1198 | 5516 |
| *For the Box (IQR and Median)* | | |
| Q2-Q1 | 1 | 0 |
| Q3-Q2 | 2 | 5 |
| *For the Whiskers* | |  |
| Q3+1.5\*IQR | 16,5 | 52,5 |
| Q1-1.5\*IQR | 4,5 | 32,5 |
| Upper Whisker | 16 | 52,5 |
| Lower Whisker | 4,5 | 32,5 |
| Wupper-Q3 | 4 | 7,5 |
| Q1-Wlower | 4,5 | 7,5 |
| *For the Outliers* | |  |
| Max | #N/A | 99 |
| Min | 1 | 1 |
|  |  |  |

We can deduce from the Boxplots that there are Minima and Maxima located largely outside the 50% of the data. “Hours-per-week” gives us an example of a complete Boxplot, having also a Max and Min Outlier.

Outliers will be treated by applying discretization, letting the filter chose the appropriate number of bins.

As said before, around 7% of instances which have one or more missing values. We have chosen to replace these missing values by a mean value, in order to keep also those instances.

# Similarity computation techniques relevant to your datasets

1. Blood transfusion

Since all values (except the binary value for doing transfusion in march 07) in this Dataset are numeric ones, to compare on similarity we can calculate the distance using the Minkowski distance, by first cleaning up the data by bringing attributes to a unit-less form.

1. Census-Income

Here we have in addition to numeric values also nominal values. In this case we can do a simple matching or we could create a binary mapping for those values

# Data mining tasks that you think are relevant to the datasets

1. Blood transfusion

For this Dataset we will introduce a class BIG AND SMALL DONATOR, in order to know if someone is a influent donator or not. So we can build a classifier:

We stated this convention:

If TIME(time since first donation)>49 & FREQ(freq. of donation)>24 THEN BIG

If TIME(time since first donation)<=49 & FREQ(freq. of donation)>11 THEN BIG

1. Census-Income

This Dataset gives us more fields to play with. In fact, we can create classifiers, decision trees make predictions etc. upon all attributes, pointing to the income class for instance. For example we could construct a tree which tells us if a 20-24 or a 25-30 with different work class can be mapped in a >50K income class.

E.G Age + Work 🡺 Class income: >50K /<=50K

25-30

20-24

Yes

No

Yes

No

# Give a very brief description of the different classification algorithms mentioning their key ideas and their fundamental differences

## Define the important parameters of each algorithm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Decision tree | Naïve Bayesian Classifier | Logistic Regression | KNN classifier |
| Input Parameter | The attributes of a tuple | The posterior probabilities of Hypothesis H based on additional information | The measure of the total contribution of all independent variables used in the model. | A new unknown tuple for which a class has to be assigned. |
| Principle | The attributes of a tuple are tested against the decision tree and a path is traced from the root to a leaf node which holds the  prediction for that tuple | Given a tuple X, the classifier will predict that X belongs to the  class having the highest posterior probability conditioned on X. | Given x representing the exposure to  some set of risk factors, LR predicts the probability of occurrence of an event by fitting  data to a logistic curve, f(x), which represents the probability of  a particular outcome | Nearest-neighbor classifiers compare a given test tuple with  training tuples that are similar and described by n attributes and are stored in n-dimensional space  🡺 Find the k-nearest tuples from the training set to the unknown tuple |
| ALG. FORMULA | At start, all the training tuples are at the root. Then, tuples are partitioned recursively based on selected attributes. The test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain). We stop when all samples for a given node belong to the same class or there are no remaining attributes for further partitioning 🡺 majority voting. | At start, compute P(C) The prior probability of. Each class can be computed based on the training tuples. Then compute each independent probability for attribute xi in reference with Class C and multiply P (|C). Example: The probability of class C to be yes is P(C) = 9/14. The amount of attribute Xi being test with respect to C is P(|) =  Finally, compute P(X|)P() for each class 🡺 The naïve Bayesian Classifier predicts C=yes for tuple X | Since x is defined as x = the LR is  Estimate the parameters using the Maximum Likelihood Function and then by computing the partial derivatives of the log likelihood, equate each partial derivative to zero, and solve the resulting nonlinear equations | The ALG assigns for the unknown tuple the most common class among its k-nearest neighbor. When k=1 the unknown tuple is assigned the class of the training tuple that is closest to it. To measure the distance we can use the Euclidean distance  **Choose K:** If k=1 the classification will be 1:1 sensitive to other data.  If k=n we’ll suffer high noise.  Go through various K’s and choose one giving lowest misclassification error! |

# Describe the setting of the experiments and the approach you have followed to analyze the behavior of the classification algorithms. (Training sets, test sets, evaluation methodology, measures)

1. Blood transfusion

For this dataset we removed the “donated in march” attributes, which is pointless for our usage.

We created a training set containing the new class BIG/SMALL DONOR using around 4% of the tuples. (4% of 748 approx = 32 tuples)

The class was leaved empty on the test set, which is the entire dataset.

The classifiers where firstly executed on the training sets, and then applied to the tests sets, in order to get a clean prediction.

1. Census-Income

Since this dataset has a lot of attributes, and for instance fnlwgt seems to be useless, we removed some based on the Information Gain Technique. This should speed up efficiency.

The *Information Gain* measure ranked the attributes (in order of importance): 8,6, 7, 1, 4, 5, 11, 13, 10, 12, 2, 14, 9 ,3

So we focused only on this attributes: *age(a1), education(a2), marital-status(a3), relationship(a4), capital-gain(a5), capitalloss(a6)* and *hours-per-week(a7)*.

The classifiers where firstly executed on the training sets, and then applied to the tests sets, in order to get a clean prediction.

# Describe the results and provide explanations and comparative study of the algorithms and the datasets.

1. Blood transfusion

## Decision Tree

|  |  |
| --- | --- |
| On training set | On test set |
| Correctly Classified Instances 31 96.875 %  Incorrectly Classified Instances 1 3.125 %  === Detailed Accuracy By Class ===  TP Rate FP Rate Precision Recall F-Measure ROC Area Class  1 0.077 0.95 1 0.974 0.966 BIG  0.923 0 1 0.923 0.96 0.966 SMALL  Weighted Avg. 0.969 0.046 0.97 0.969 0.969 0.966 | **Correctly Classified Instances 748 100 %**  **Incorrectly Classified Instances 0 0 %**  **=== Detailed Accuracy By Class ===**  **TP Rate FP Rate Precision Recall F-Measure ROC Area Class**  **1 0 1 1 1 1 BIG**  **1 0 1 1 1 1 SMALL**  **Weighted Avg. 1 0 1 1 1 1** |

## Naïve Bayesian Classifier

|  |  |
| --- | --- |
| On training set | On test set |
| Correctly Classified Instances: 25 78.125 %  Incorrectly Classified Instances: 7 21.875 %  === Detailed Accuracy By Class ===  TP Rate FP Rate Precision Recall F-Measure ROC Area Class  0.684 0.077 0.929 0.684 0.788 0.872 BIG  0.923 0.316 0.667 0.923 0.774 0.872 SMALL  Weighted Avg. 0.781 0.174 0.822 0.781 0.782 0.872  === Confusion Matrix ===  a b <-- classified as  13 6 | a = BIG  1 12 | b = SMALL | **Correctly Classified Instances: 719 96.123 %**  **Incorrectly Classified Instances: 29 3.877 %**  **=== Detailed Accuracy By Class ===**  **TP Rate FP Rate Precision Recall F-Measure ROC Area Class**  **0.969 0.118 0.988 0.969 0.978 0.981 BIG**  **0.882 0.031 0.741 0.882 0.805 0.981 SMALL**  **Weighted Avg. 0.961 0.11 0.966 0.961 0.963 0.981**  **=== Confusion Matrix ===**  **a b <-- classified as**  **659 21 | a = BIG**  **8 60 | b = SMALL** |

## Logistic Regression

|  |  |
| --- | --- |
| On training set | On test set |
| Correctly Classified Instances 29 90.625 %  Incorrectly Classified Instances 3 9.375 %  === Detailed Accuracy By Class ===  TP Rate FP Rate Precision Recall F-Measure ROC Area Class  0.947 0.154 0.9 0.947 0.923 0.907 BIG  0.846 0.053 0.917 0.846 0.88 0.907 SMALL  Weighted Avg. 0.906 0.113 0.907 0.906 0.906 0.907  === Confusion Matrix ===  a b <-- classified as  18 1 | a = BIG  2 11 | b = SMALL | **Correctly Classified Instances 746 99.7326 %**  **Incorrectly Classified Instances 2 0.2674 %**  **=== Detailed Accuracy By Class ===**  **TP Rate FP Rate Precision Recall F-Measure ROC Area Class**  **0.993 0 1 0.993 0.996 1 BIG**  **1 0.007 0.996 1 0.998 1 SMALL**  **Weighted Avg. 0.997 0.005 0.997 0.997 0.997 1**  **=== Confusion Matrix ===**  **a b <-- classified as**  **267 2 | a = BIG**  **0 479 | b = SMALL** |

## KNN Classifier (KNN = 1)

|  |  |
| --- | --- |
| On training set | On test set |
| Correctly Classified Instances 32 100 %  Incorrectly Classified Instances 0 0 %  === Confusion Matrix ===  a b <-- classified as  19 0 | a = BIG  0 13 | b = SMALL | **Correctly Classified Instances 745 99.5989 %**  **Incorrectly Classified Instances 3 0.4011 %**  **=== Confusion Matrix ===**  **a b <-- classified as**  **579 2 | a = BIG**  **1 166 | b = SMALL** |

## KNN Classifier (KNN = 3)

|  |  |
| --- | --- |
| On training set | On test set |
| Correctly Classified Instances 30 93.75 %  Incorrectly Classified Instances 2 6.25 %  === Confusion Matrix ===  a b <-- classified as  19 0 | a = BIG  2 11 | b = SMALL | **Correctly Classified Instances 741 99.0642 %**  **Incorrectly Classified Instances 7 0.9358 %**  **=== Confusion Matrix ===**  **a b <-- classified as**  **642 2 | a = BIG**  **5 99 | b = SMALL** |

## KNN Classifier (KNN = 5)

|  |  |
| --- | --- |
| On training set | On test set |
| Correctly Classified Instances 26 81.25 %  Incorrectly Classified Instances 6 18.75 %  === Confusion Matrix ===  a b <-- classified as  17 2 | a = BIG  4 9 | b = SMALL | **Correctly Classified Instances 741 99.0642 %**  **Incorrectly Classified Instances 7 0.9358 %**  **=== Confusion Matrix ===**  **a b <-- classified as**  **632 2 | a = BIG**  **5 109 | b = SMALL** |

1. Census-Income

## Decision tree

The tree was tested on the full dataset (with *fnlwgt* and *education-num* attributes removed and missing values replaced).

**Complexity Control Parameter Value Accuracy**

Post-pruning 0.35 85.90

Post-pruning 0.30 85.88

**Post-pruning 0.25 86.02**

**Post-pruning 0.20 86.02**

Post-pruning 0.15 85.98

Post-pruning 0.10 85.78

Minimum number of objects 10 85.85

Minimum number of objects 20 85.77

Minimum number of objects 30 85.75

The results show that post-pruning yields he best results, with a very modest 0.20 setting.

This is the highest value this model has generalised too, at **86.02%.**

## Naïve Bayesian Classifier

The model was tested on a the dataset (with *fnlwgt* and *educationnum* attributes removed and missing values replaced). This yielded an accuracy of **82.31%**.

One issue that affected Naïve Bayes was that numeric values. Some were not Gaussian

in appearance, such as age, which shows a positive skew. For this phase, we applied

discretization on the model. Attributes with any different values (*age*) were discretized

into 10 bins, while those with few values (*capital-loss* & *capital-*gain) were discretized into two. The accuracy fell to **81.09%**.

## Logistic Regression

Logistics was tested on the sample subset (with *fnlwgt* and *education-num* attributes

removed and missing values replaced), as this yielded the models highest accuracy. We

repeated the experiments, this time altering the ridge parameter. The results were noted:

**Ridge Parameter Accuracy**

1 x 10-8 85.74

1 x 10-4 85.74

**1 85.81**

10 85.59

20 85.47

30 85.19

50 84.95

100 84.95

Logistics with a ridge parameter of 1 yielded the highest accuracy yet for this model at **85.81%**

## KNN Classifier

So far, this models strongest performance (83.20%) was on the sample subset of 10%

with only 7 selected attributes and a *k* of 10. We further investigated this by applying the model again on the same sample subset and altering the models weightings. This allows you to adapt the influence of the neighbours according to their distance. The results are recorded as follows:

**Weighting Accuracy**

Standard 83.20

1/distance 83.10

**1-distance 83.93**

1-distance demonstrated a record accuracy for this model at **83.93%**.