APPLIED MACHINE LEARNING

(MINI PROJECT)

Group Members

Aditya Elkunchwar (aelkunch)

Rituraj Singh (risingh)

Himanshu Goyal(hgoyal)

Table of Contents

1 Project Overview

In this mini-project – we analyzed 3 UCI datasets – using Weka's inbuilt classification algorithms. The datasets used are:

- German Credit Card
- Adult
- Hand Written Digits

The objective, was to **run**, **interpret** and **compare** the results of the following **5 algorithms**:

- Naive Bayes
- Logistic Regression
- J48
- Random Forest
- ZeroR

Train-Test-Set-Splits
We used two strategies to split Training and Test sets:

- 70% - Train and 30% - Test
- 10 fold cross validation.

Comparison Strategy
We recorded the Confusion Matrices and other metrics (AUC ROC, Accuracy, Error Rate) for each combination – using the two test strategies mentioned above. Finally - We **picked the best results** for every Algorithm - and based our interpretation on those results. The same exercise was repeated for all 3 datasets.

Our understanding of algorithms and observations on different datasets are appended below.

2 Overview of Algorithms (our takeaway)

2.1 Naive Bayes

Naive Bayes is a collection of classification algorithms based on Bayes Theorem – which assumes that, every feature being classified is independent of the value of any other feature. The algorithm allows us to predict a class, given a set of features using probability.

The way its works, is that – based on the training set, the algorithm already calculates the probability of each features – given that the output is known. It can then take a new input – and predict the output based on the probabilities it already knows. The algorithm works with Nominal as well as Numeric values.

Pros

- It's relatively simple to understand and build
- It's easily trained, even with a small dataset
- It's fast!
- It's not sensitive to irrelevant features

Cons

- It assumes every feature is independent, which isn't always the case

2.2 Logistic Regression

Logistic regression is the go-to method for binary classification problems. It is named for the function used at the core of the method - the logistic (or sigmoid) function. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

Logistic regression is represented using a linear equation. The input values (x) are combined linearly using weights or coefficient values to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

The coefficients of the logistic regression algorithm are estimated from training data using maximum-likelihood estimation. The best coefficients results in a model that would predict a value very close to 1 for the default class and a value very close to 0 for the other class. Any new prediction can be made as per equation - where the coefficients and input values are known.

2.3 Decision Tree

Decision tree is one of the most popular classification algorithms used in problems. It is supervised learning (possessing pre-defined target values) in nature, it can work in both categorical and continuous inputs and variable outputs. Decision tree works on the principal of splitting the data set into two or more homogeneous sets.

There are two types of decision trees they are

- Categorical variable decision tree
- Continuous variable decision tree

The most important activity in decision tree is to identify the root node, splitting the data set into homogeneous sets. The pros and cons of using a decision tree are listed below.

Pros

- Easy to understand
- Useful in data exploration
- Less data cleaning required
- Data type is not a constraint
- Non parametric method

Cons

- Over fitting – Decision trees can easily overfit if enough instances are given.
- Not a good fit for continuous variables (although it is possible).

2.4 Random Forests

Random forest is considered to be a cure-all solution for data science problems. It is highly favored due to its versatility as it is capable of performing both classification and regression tasks. Random forest can be defined as the combination of several weak models providing a strong model. Multiple trees are grown. In case of classification on a new attribute, each tree provides it classification and the forest "decides" the classification that has most number of entries.

Pros

- Can solve both classical and regression problems
- It can handle larger data sets with high dimensionality
- Its efficiency in handling missing data and predicting the values of those data
- The efficient way of balancing errors on data sets
- The usage of bootstrap sampling

Cons

- The models work great for classification problem than regression problems
- It is a black box approach for statistical modelers

2.5 ZeroR

Zero R – simply stated – is a majority class classifier.

Although there is no predictability power in ZeroR, it is useful for determining a baseline performance as a benchmark for other classification methods.

We have included this algorithm in our report – purely for comparison purposes. Virtually any prediction methodology that we use – SHOULD be an improvement over ZeroR results to be acceptable.

3 German Credit Card data

3.1 Data Source

UC Irvine Machine Learning Repository
Professor Dr. Hans Hofmann
Institut f"ur Statistik und "Okonometrie
Universit"at Hamburg
FB Wirtschaftswissenschaften
Von-Melle-Park 5
2000 Hamburg 13

3.2 Data Description

The data set contains 1000 customers' information about bank account, credit history and present credit situation, credit purpose, property and installment, age, sex, personal status, employment, residence as its 20 independent variables, as well as that whether they are evaluated by experts a good customer with low

credit risk or bad customer with high credit risk as its response. The purpose of analyzing this dataset is to predict a customer's credit risk based on the other information of the customer. Since wrongly predicting a true bad customer as a good one is much worse than wrongly predicting a true good customer as a bad one.

3.3 Classification Algorithm Results

3.3.1 Naive Bayes

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     GermanCreditData-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21
Instances:    1000
Attributes:   21
              CurAcctBal
              DurationMonth
              PrevPayments
              CreditPurpose
              DMCreditAmt
              SavingsVal
              EmpDuration
              rate
              MaritalStatus
              Gaurantors
              StayDuration
              MostValAsset
              Age
              FurtherCredits
              ApptType
              PrevCreditCount
              Occupation
              PersonCountMaint
              Telephone
              ForeignWorker
               Creditability
Test mode:    10-fold cross-validation


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         753               75.3    %
Incorrectly Classified Instances       247               24.7    %
Kappa statistic                          0.3794
Mean absolute error                      0.2949
Root mean squared error                  0.4229
Relative absolute error                 70.1861 %
Root relative squared error             92.2863 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.497    0.137    0.608      0.497    0.547      0.383    0.781     0.571     0
                 0.863    0.503    0.800      0.863    0.830      0.383    0.781     0.884     1
Weighted Avg.    0.753    0.393    0.742      0.753    0.745      0.383    0.781     0.790

=== Confusion Matrix ===

   a   b   <-- classified as
 149 151 |   a = 0
  96 604 |   b = 1
```

3.3.2 Logistic Regression

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     GermanCreditData-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21
Instances:    1000
Attributes:   21
              CurAcctBal
              DurationMonth
              PrevPayments
              CreditPurpose
              DMCreditAmt
              SavingsVal
              EmpDuration
              rate
              MaritalStatus
              Gaurantors
              StayDuration
              MostValAsset
              Age
              FurtherCredits
              ApptType
              PrevCreditCount
              Occupation
              PersonCountMaint
              Telephone
              ForeignWorker
               Creditability
Test mode:    10-fold cross-validation


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         751               75.1    %
Incorrectly Classified Instances       249               24.9    %
Kappa statistic                          0.3731
Mean absolute error                      0.3124
Root mean squared error                  0.4123
Relative absolute error                 74.3384 %
Root relative squared error             89.9757 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.490    0.137    0.605      0.490    0.541      0.377    0.774     0.602     0
                 0.863    0.510    0.798      0.863    0.829      0.377    0.774     0.868     1
Weighted Avg.    0.751    0.398    0.740      0.751    0.743      0.377    0.774     0.788

=== Confusion Matrix ===

   a   b   <-- classified as
 147 153 |   a = 0
  96 604 |   b = 1
```

---

1) Odds ratios for attributes with greater than 1 can be classified as good. For example, when curAcctBal is 1, 2, we have odds ratios as 2.2091 and 1.5133 respectively which is consistent with DMCreditAmt.
2) We have the best Coefficient values when CurAcctBal is 1 and 2.

3.3.3 J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     GermanCreditData-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21
Instances:    1000
Attributes:   21
              CurAcctBal
              DurationMonth
              PrevPayments
              CreditPurpose
              DMCreditAmt
              SavingsVal
              EmpDuration
              rate
              MaritalStatus
              Gaurantors
              StayDuration
              MostValAsset
              Age
              FurtherCredits
              ApptType
              PrevCreditCount
              Occupation
              PersonCountMaint
              Telephone
              ForeignWorker
               Creditability
Test mode:    10-fold cross-validation


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         729               72.9    %
Incorrectly Classified Instances       271               27.1    %
Kappa statistic                          0.3037
Mean absolute error                      0.3294
Root mean squared error                  0.4542
Relative absolute error                 78.3941 %
Root relative squared error             99.1212 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.420    0.139    0.565      0.420    0.482      0.310    0.677     0.499     0
                 0.861    0.580    0.776      0.861    0.817      0.310    0.677     0.769     1
Weighted Avg.    0.729    0.448    0.713      0.729    0.716      0.310    0.677     0.688

=== Confusion Matrix ===

   a   b   <-- classified as
 126 174 |   a = 0
  97 603 |   b = 1
```

1) This algorithm selected CurAcctBal, DurationMonth, and Occupation as top 3 attributes.
2) If CurAcctBal is 3 or 4, classify customer as Bad.
3) If CurAcctBal is 1, what is DurationMoth.
4) If CurAcctBal is 2, what is DMCreditAmt.

3.3.4 Random Forest

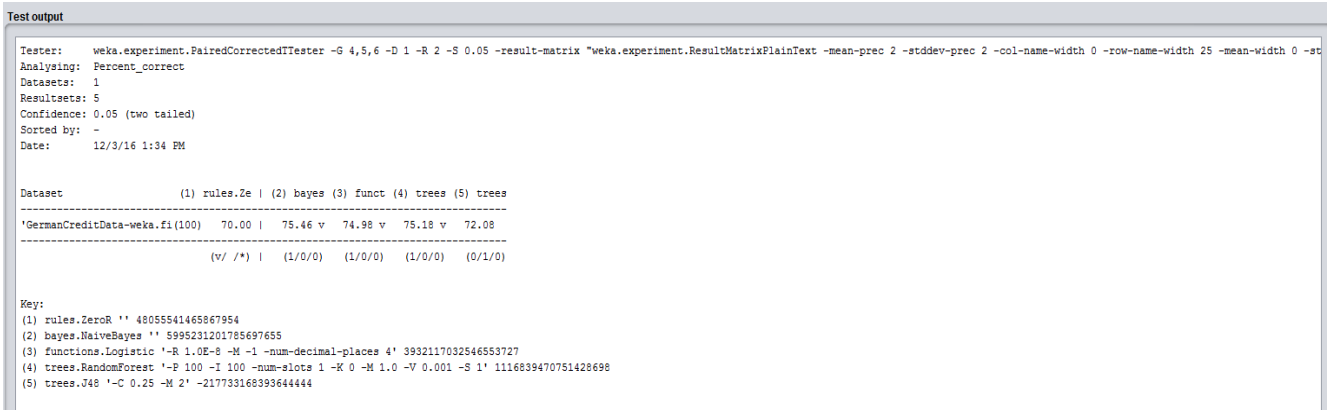
```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     GermanCreditData-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21
Instances:    1000
Attributes:   21
              CurAcctBal
              DurationMonth
              PrevPayments
              CreditPurpose
              DMCreditAmt
              SavingsVal
              EmpDuration
              rate
              MaritalStatus
              Gaurantors
              StayDuration
              MostValAsset
              Age
              FurtherCredits
              ApptType
              PrevCreditCount
              Occupation
              PersonCountMaint
              Telephone
              ForeignWorker
               Creditability
Test mode:    10-fold cross-validation


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         750               75      %
Incorrectly Classified Instances       250               25      %
Kappa statistic                          0.3294
Mean absolute error                      0.3469
Root mean squared error                  0.4115
Relative absolute error                 82.554  %
Root relative squared error             89.794  %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.387    0.094    0.637      0.387    0.481      0.347    0.769     0.601     0
                 0.906    0.613    0.775      0.906    0.835      0.347    0.769     0.875     1
Weighted Avg.    0.750    0.458    0.734      0.750    0.729      0.347    0.769     0.793

=== Confusion Matrix ===

   a   b   <-- classified as
 116 184 |   a = 0
  66 634 |   b = 1
```

1) On the contrary Random forest selected Credit Purpose, Duration Month and Age as top 3 attributes.

3.3.5 ZeroR

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     GermanCreditData-weka.filters.unsupervised.attribute.NumericToNominal-R1,3,4,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21
Instances:    1000
Attributes:   21
              CurAcctBal
              DurationMonth
              PrevPayments
              CreditPurpose
              DMCreditAmt
              SavingsVal
              EmpDuration
              rate
              MaritalStatus
              Gaurantors
              StayDuration
              MostValAsset
              Age
              FurtherCredits
              ApptType
              PrevCreditCount
              Occupation
              PersonCountMaint
              Telephone
              ForeignWorker
               Creditability
Test mode:    10-fold cross-validation


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         700               70      %
Incorrectly Classified Instances       300               30      %
Kappa statistic                          0
Mean absolute error                      0.4202
Root mean squared error                  0.4583
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.300     0
                 1.000    1.000    0.700      1.000    0.824      0.000    0.500     0.700     1
Weighted Avg.    0.700    0.700    0.490      0.700    0.576      0.000    0.500     0.580

=== Confusion Matrix ===

   a   b   <-- classified as
   0 300 |   a = 0
   0 700 |   b = 1
```

3.4 Comparison and interpretation of Results



Test output

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width 0 -row-name-width 25 -mean-width 0 -st
Analysing:  Percent_correct
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/3/16 1:34 PM


Dataset                   (1) rules.Ze | (2) bayes (3) funct (4) trees (5) trees
----------------------------------------------------------------------------------
'GermanCreditData-weka.fi(100)    70.00 |   75.46 v   74.98 v   75.18 v   72.08
----------------------------------------------------------------------------------
                              (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (0/1/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(5) trees.J48 '-C 0.25 -M 2' -217733168393644444
```

On comparing the results, it was found that Naïve Bayes had the best performance with accuracy of 75.46% followed by random forest 75.18%, logistic regression 74.98%, J48, zero R.

The reason Naïve Bayes performed well because it is known to perform well on relatively small data. Also, it handles the irrelevant features well.

1) Naïve Bayes, Random Forest and Logistic regression gave nearly the same results. we cannot simply tell that which model gives better result.
2) Precision is more important in this problem as compared to Accuracy and Recall as we should be ok in missing a good customer but we should not be approving a bad customer.
3) Naïve Bayes has better precision as compared to other 4 algorithms.
4) Performance of decision tree was not good because features cannot be differentiated into category. We can see that Random forest also struggled a bit because of the same reason. This can be justified by selection of the top 3 attributes. J48 picks CurAcctBal, DurationMonth, Occupation as top 3 attributes whereas random forest picks CreditPurpose, DurationMonth and Age as top 3 attributes.
5) Naïve Bayes and logistic regression as both based on probability and seems to perform better than rule base algorithms like decision tree and random forest.

4 Adult data

4.1 Data Source

UC Irvine Machine Learning Repository
Donor:
Ronny Kohavi and Barry Becker
Data Mining and Visualization
Silicon Graphics.
e-mail: ronnyk '@' live.com for questions.

4.2 Data Description

This data was extracted from the census bureau database found at http://www.census.gov/ftp/pub/DES/www/welcome.html. The instances are a mix of continuous and discrete values for Adults. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0)): **48842** instances.

Prediction task is to determine whether a person makes over 50K a year.

4.3 Classification Algorithm Results

4.3.1 Naive Bayes

In this dataset, the algorithm trains on the dataset – knowing if the Adult makes >50 K or less. It calculates probabilities of the individual features. Our dataset has 15 features. So it learns the individual feature probabilities – given the "result" i.e. 50K or more / less.
In question form:
Given that a Person makes > 50K - what is the probability of person being "**white**" ?
Given that a Person makes > 50K - what is the probability of person being "**man**" ?
Given that a Person makes <50K - what is the probability of person's country being "**'US**" ?
Given that a Person make < 50K - what is the probability of person having a "**bachelors degree**" ?
Given that a Person makes > 50K - what is the probability of person's age being **40** ?

The Results are as follows:

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     adult_data
Instances:    48842
Attributes:   15
              age
              workclass
              fnlwgt
              education
              education-num
              marital-status
              occupation
              relationship
              race
              sex
              capital-gain
              capital-loss
              hours-per-week
              native-country
              ClassResult
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       40693               83.3156 %
Incorrectly Classified Instances      8149               16.6844 %
Kappa statistic                          0.4953
Mean absolute error                      0.174
Root mean squared error                  0.3732
Relative absolute error                 47.8065 %
Root relative squared error             87.475  %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.933    0.483    0.860      0.933    0.895      0.505    0.892     0.964      <=50K
                 0.517    0.067    0.707      0.517    0.597      0.505    0.892     0.723      >50K
Weighted Avg.    0.833    0.383    0.823      0.833    0.824      0.505    0.892     0.906

=== Confusion Matrix ===

     a     b   <-- classified as
 34649  2506 |     a =  <=50K
  5643  6044 |     b =  >50K
```

4.3.2 Logistic Regression

As mentioned in the overview, Logistic Regression calculates the coefficients based on maximum likely estimation. These coefficients help predict the binary class (as close to one or the other) by giving out probability. The attributes such age, race, sex, education – each have a positive or negative coefficients – that contribute to the overall prediction. The Weka output also shows the odds ratio for each feature.

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     adult_data
Instances:    48842
Attributes:   15
              age
              workclass
              fnlwgt
              education
              education-num
              marital-status
              occupation
              relationship
              race
              sex
              capital-gain
              capital-loss
              hours-per-week
              native-country
              ClassResult
Test mode:    10-fold cross-validation
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       41633               85.2402 %
Incorrectly Classified Instances      7209               14.7598 %
Kappa statistic                          0.5676
Mean absolute error                      0.202
Root mean squared error                  0.3191
Relative absolute error                 55.4907 %
Root relative squared error             74.7977 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.932    0.400    0.881      0.932    0.906      0.572    0.906     0.968      <=50K
                 0.600    0.068    0.734      0.600    0.661      0.572    0.906     0.766      >50K
Weighted Avg.    0.852    0.320    0.846      0.852    0.847      0.572    0.906     0.919

=== Confusion Matrix ===

     a     b   <-- classified as
 34615  2540 |     a =  <=50K
  4669  7018 |     b =  >50K
```

4.3.3 J48 (Decision Tree)

In decision tree, splitting based on its entropy or effectively, the Information you can 'gain' from the attribute. In this dataset, capital-gain, marital-status and education-num – were the highest level nodes – based on which splitting happened.

The Results are as follows:

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     adult_data
Instances:    48842
Attributes:   15
              age
              workclass
              fnlwgt
              education
              education-num
              marital-status
              occupation
              relationship
              race
              sex
              capital-gain
              capital-loss
              hours-per-week
              native-country
              ClassResult
Test mode:    10-fold cross-validation
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       41987               85.9649 %
Incorrectly Classified Instances      6855               14.0351 %
Kappa statistic                          0.5851
Mean absolute error                      0.1939
Root mean squared error                  0.3214
Relative absolute error                 53.2542 %
Root relative squared error             75.3278 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.940    0.396    0.883      0.940    0.911      0.591    0.887     0.946      <=50K
                 0.604    0.060    0.760      0.604    0.673      0.591    0.887     0.744      >50K
Weighted Avg.    0.860    0.316    0.854      0.860    0.854      0.591    0.887     0.897

=== Confusion Matrix ===

     a     b   <-- classified as
 34933  2222 |     a =  <=50K
  4633  7054 |     b =  >50K
```

4.3.4 Random Forest

Our data set has 48842 instances. Using this technique, the algorithm created 100 bags (of randomly selected instances). Then decision trees were built on each of these 100 bags. For creating a final classification, the features were selected "based on highest number of elements – they split" – i.e. selecting the best classifier out of all.

```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     adult_data
Instances:    48842
Attributes:   15
              age
              workclass
              fnlwgt
              education
              education-num
              marital-status
              occupation
              relationship
              race
              sex
              capital-gain
              capital-loss
              hours-per-week
              native-country
              ClassResult
Test mode:    10-fold cross-validation
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       41489               84.9453 %
Incorrectly Classified Instances      7353               15.0547 %
Kappa statistic                          0.567
Mean absolute error                      0.1949
Root mean squared error                  0.326
Relative absolute error                 53.5471 %
Root relative squared error             76.4191 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.922    0.380    0.885      0.922    0.903      0.569    0.896     0.962      <=50K
                 0.620    0.078    0.713      0.620    0.663      0.569    0.896     0.753      >50K
Weighted Avg.    0.849    0.308    0.844      0.849    0.846      0.569    0.896     0.912

=== Confusion Matrix ===

     a     b   <-- classified as
 34244  2911 |     a =  <=50K
  4442  7245 |     b =  >50K
```

4.3.5 ZeroR

We have included this algorithm in our report – purely for comparison purposes. Virtually any prediction methodology that we use – SHOULD be an improvement over ZeroR results to be acceptable.

For a Binary classifier – (like we have in this dataset) - ZeroR simply looks at the Frequency of "Adults with income 50K or higher" and Frequency if "Adults with income lower than 50K". Depending on whichever is greater – It predicts rest all inputs (from test set) – as the majority class.

The Results are as follows:

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     adult_data
Instances:    48842
Attributes:   15
              age
              workclass
              fnlwgt
              education
              education-num
              marital-status
              occupation
              relationship
              race
              sex
              capital-gain
              capital-loss
              hours-per-week
              native-country
              ClassResult
Test mode:    10-fold cross-validation
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       37155               76.0718 %
Incorrectly Classified Instances     11687               23.9282 %
Kappa statistic                          0
Mean absolute error                      0.3641
Root mean squared error                  0.4266
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    1.000    0.761      1.000    0.864      0.000    0.500     0.761      <=50K
               0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.239      >50K
Weighted Avg.  0.761    0.761    0.579      0.761    0.657      0.000    0.500     0.636

=== Confusion Matrix ===

     a     b   <-- classified as
 37155     0 |     a =  <=50K
 11687     0 |     b =  >50K
```

4.4 Comparison and interpretation of Results

Weka experimenter allows us to conduct a T-test of any of the output metrics and gives us a comparison of the performance of each algorithm.

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrix
Analysing:  Percent_correct
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/8/16 12:13 AM


Dataset                  (1) rules.Ze | (2) bayes (3) funct (4) trees (5) trees
-------------------------------------------------------------------------------
adult_data               (25)   76.07 |   83.33 v   85.25 v   85.98 v   84.82 v
-------------------------------------------------------------------------------
                              (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
```

On comparing these algorithms – on the dataset – we found that almost all the algorithms performed very well. The J48 - Decision Tree algorithm was best with accuracy of 85.98%. The other good performers were – Logistic Regression –with 85.25 % and Random Forest – with 84.82%

Here is why we think the algorithms performed the way – they did:

Decision Tree – The data followed a pattern – where it could be correctly classified in a tree structure. The features were able to provide clear differentiation – which helped in deciding the root (and parent) nodes. There was also a large number of instances to train on ~48K. We used 10-fold test as well. Here we cannot ignore the possibility that the decision tree may have over fit – but until we test against new data, we wouldn't know. For now – we take the accuracy on face value.

Random forest - We see that accuracy of random forest was slightly lower than the decision tree – but still good. This could be because – random forest used 100 bags of randomly selected instances – which caused a slight deviation in "decision making". The Decision Tree had "all of the instances" to correctly calculate information gain (and /or minimum entropy) to select root (and parent) nodes. Random forests "normalized " its decisions (which is probably more reliable) – but decision tree outperformed it.

Naïve Bayes was the lowest performing algorithm (other than ZeroR). It assumed that all the features were independent of each other – however we believe many features "collectively" influence income levels. For example – Education-Level and Years-of-Education are related. Workclass and Hours of work are also somewhat related. (People in Private sector – worked significantly higher hours).

Logistic regression – performed fairly well – because we had a lot of instances ~48 thousand instances. This gave a good mathematical base to use maximum likelihood estimation. The coefficients were finely calculated and hence sigmoid function was able to give best binary classification.

As, mentioned earlier, we included results from ZeroR – just as a baseline, to show that virtually any algorithm is significantly better that it.

More Weka comparisons (RMSE):

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrix
Analysing:  Root_mean_squared_error
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/8/16 12:14 AM


Dataset                   (1) rules.Z | (2) baye (3) func (4) tree (5) tree
----------------------------------------------------------------------------
adult_data                (25)   0.43 |   0.37 *   0.32 *   0.32 *   0.33 *
----------------------------------------------------------------------------
                              (v/ /*) |  (0/0/1)  (0/0/1)  (0/0/1)  (0/0/1)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
```

Decision Tree and Logistic Regression are top performers. (Lowest RMSE).

More Weka comparisons (Area under ROC):

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrix
Analysing:  Area_under_ROC
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/8/16 12:16 AM


Dataset                   (1) rules.Z | (2) baye (3) func (4) tree (5) tree
----------------------------------------------------------------------------
adult_data                (25)   0.50 |   0.89 v   0.91 v   0.88 v   0.90 v
----------------------------------------------------------------------------
                              (v/ /*) |  (1/0/0)  (1/0/0)  (1/0/0)  (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
```

Logistic Regression and Random Forest are top performers.

More Weka comparisons (Precision):

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrix
Analysing:  IR_precision
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/8/16 12:17 AM


Dataset                   (1) rules.Z | (2) baye (3) func (4) tree (5) tree
----------------------------------------------------------------------------
adult_data                (25)   0.76 |   0.86 v   0.88 v   0.88 v   0.88 v
----------------------------------------------------------------------------
                              (v/ /*) |  (1/0/0)  (1/0/0)  (1/0/0)  (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
```

Decision trees, Logistic Regression and Random Forest are equally good.

More Weka comparisons (Recall):

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrix
Analysing:  IR_recall
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/8/16 12:18 AM


Dataset                   (1) rules.Z | (2) baye (3) func (4) tree (5) tree
----------------------------------------------------------------------------
adult_data                (25)   1.00 |   0.93 *   0.93 *   0.94 *   0.92 *
----------------------------------------------------------------------------
                              (v/ /*) |  (0/0/1)  (0/0/1)  (0/0/1)  (0/0/1)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
```

Decision Tree and Logistic Regression are top performers.

5 Hand written digits data

5.1 Data Source

Title: Semeion Handwritten Digit
Abstract: 1593 handwritten digits from around 80 persons were scanned,
stretched in a rectangular box 16x16 in a gray scale of 256 values.

Data Set Characteristics: Multivariate
Number of Instances: 1593
Area: Computer
Attribute Characteristics: Integer
Number of Attributes: 256
Date Donated: 2008-11-11
Associated Tasks: Classification
Missing Values? N/A

Source:
The dataset was created by Tactile Srl, Brescia, Italy
(http://www.tattile.it/) and donated in 1994 to Semeion Research Center of Sciences of Communication, Rome, Italy (http://www.semeion.it/), for machine learning research.

5.2 Data Description

1593 handwritten digits from around 80 persons were scanned, stretched in a rectangular box 16x16 in a gray scale of 256 values. Then each pixel of each image was scaled into a boolean (1/0) value using a fixed threshold.

Each person wrote on a paper all the digits from 0 to 9, twice. The commitment was to write the digit the first time in the normal way (trying to write each digit accurately) and the second time in a fast way (with no accuracy).

5.3 Classification Algorithm Results

5.3.1 Naive Bayes

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     semeion1-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    1593
Attributes:   257
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1358               85.248  %
Incorrectly Classified Instances       235               14.752  %
Kappa statistic                          0.8361
Mean absolute error                      0.0297
Root mean squared error                  0.1653
Relative absolute error                 16.5019 %
Root relative squared error             55.1068 %
Total Number of Instances             1593

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950     0.002     0.981       0.950   0.965      0.962    0.996     0.980     0
                 0.778     0.038     0.696       0.778   0.735      0.704    0.970     0.821     1
                 0.824     0.008     0.923       0.824   0.870      0.859    0.989     0.946     2
                 0.881     0.023     0.809       0.881   0.843      0.826    0.982     0.921     3
                 0.801     0.013     0.872       0.801   0.835      0.818    0.983     0.912     4
                 0.855     0.006     0.944       0.855   0.898      0.888    0.996     0.970     5
                 0.876     0.007     0.934       0.876   0.904      0.894    0.996     0.976     6
                 0.918     0.028     0.784       0.918   0.845      0.830    0.991     0.928     7
                 0.845     0.020     0.819       0.845   0.832      0.813    0.987     0.897     8
                 0.797     0.019     0.824       0.797   0.810      0.790    0.978     0.884     9
Weighted Avg.    0.852     0.016     0.859       0.852   0.854      0.838    0.987     0.923
```

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 153   0   1   0   2   0   0   0   5   0 |   a = 0
   0 126   5   3   3   2   1  22   0   0 |   b = 1
   0  14 131   0   4   0   2   0   5   3 |   c = 2
   0   7   0 140   0   1   0   1   4   6 |   d = 3
   0  14   1   0 129   1   3   8   0   5 |   e = 4
   0   1   1   8   2 136   3   4   1   3 |   f = 5
   3   5   1   0   5   3 141   0   3   0 |   g = 6
   0   5   0   0   2   0   1 145   1   4 |   h = 7
   0   5   2  10   0   0   0   1 131   6 |   i = 8
   0   4   0  12   1   1   0   4  10 126 |   j = 9
```

5.3.2 Logistic Regression

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     semeion1-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    1593
Attributes:   257
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1305               81.9209 %
Incorrectly Classified Instances       288               18.0791 %
Kappa statistic                          0.7991
Mean absolute error                      0.0371
Root mean squared error                  0.1831
Relative absolute error                 20.5874 %
Root relative squared error             61.0341 %
Total Number of Instances             1593

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.009    0.922      0.950    0.936      0.929    0.995     0.976     0
                 0.827    0.031    0.753      0.827    0.788      0.764    0.973     0.867     1
                 0.849    0.018    0.839      0.849    0.844      0.826    0.981     0.916     2
                 0.780    0.022    0.795      0.780    0.787      0.764    0.979     0.879     3
                 0.801    0.024    0.791      0.801    0.796      0.773    0.968     0.865     4
                 0.874    0.021    0.822      0.874    0.848      0.831    0.982     0.907     5
                 0.876    0.013    0.887      0.876    0.881      0.868    0.994     0.959     6
                 0.835    0.022    0.810      0.835    0.822      0.803    0.983     0.886     7
                 0.645    0.026    0.730      0.645    0.685      0.655    0.947     0.764     8
                 0.747    0.016    0.837      0.747    0.789      0.769    0.973     0.847     9
Weighted Avg.    0.819    0.020    0.819      0.819    0.818      0.799    0.978     0.887
```

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 153   0   0   1   1   3   0   2   1   0 |   a = 0
   0 134   4   0  12   1   0   8   3   0 |   b = 1
   1   6 135   1   3   1   2   3   6   1 |   c = 2
   0   9   2 124   2   9   1   5   4   3 |   d = 3
   5  10   0   2 129   2   5   3   2   3 |   e = 4
   0   2   1   6   0 139   1   4   2   4 |   f = 5
   1   2   1   1   4   3 141   0   8   0 |   g = 6
   2   3   1   4   8   4   2 132   1   1 |   h = 7
   3   7  11   7   1   4   7   4 100  11 |   i = 8
   1   5   6  10   3   3   0   2  10 118 |   j = 9
```

5.3.3 J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     semeion1-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    1593
Attributes:   257
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1208               75.8318 %
Incorrectly Classified Instances       385               24.1682 %
Kappa statistic                          0.7314
Mean absolute error                      0.0539
Root mean squared error                  0.2102
Relative absolute error                 29.9339 %
Root relative squared error             70.0774 %
Total Number of Instances             1593

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.932    0.010    0.909      0.932    0.920      0.911    0.960     0.864     0
               0.827    0.034    0.736      0.827    0.779      0.754    0.894     0.676     1
               0.686    0.043    0.637      0.686    0.661      0.622    0.834     0.490     2
               0.755    0.030    0.736      0.755    0.745      0.717    0.876     0.574     3
               0.720    0.024    0.773      0.720    0.746      0.719    0.870     0.619     4
               0.799    0.020    0.814      0.799    0.806      0.785    0.881     0.722     5
               0.888    0.018    0.846      0.888    0.867      0.852    0.948     0.790     6
               0.728    0.020    0.799      0.728    0.762      0.738    0.858     0.633     7
               0.594    0.034    0.652      0.594    0.622      0.584    0.765     0.403     8
               0.646    0.035    0.671      0.646    0.658      0.621    0.808     0.474     9
Weighted Avg.  0.758    0.027    0.758      0.758    0.757      0.731    0.870     0.626
```

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 150   0   1   2   2   0   1   1   2   2 |   a = 0
   1 134   7   2   5   3   1   2   5   2 |   b = 1
   0  13 109   4   5   1   4   5  14   4 |   c = 2
   1   4   7 120   0   6   0   5   6  10 |   d = 3
   2   9   3   1 116   6   7   6   5   6 |   e = 4
   1   0   1  13   6 127   3   1   0   7 |   f = 5
   2   4   5   1   1   0 143   0   5   0 |   g = 6
   2   8  11   1   7   1   5 115   2   6 |   h = 7
   2   6  22   3   2   3   5   7  92  13 |   i = 8
   4   4   5  16   6   9   0   2  10 102 |   j = 9
```

5.3.4 Random Forest

```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     semeion1-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    1593
Attributes:   257
              [list of attributes omitted]
Test mode:    10-fold cross-validation


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1494               93.7853 %
Incorrectly Classified Instances        99                6.2147 %
Kappa statistic                          0.9309
Mean absolute error                      0.075
Root mean squared error                  0.152
Relative absolute error                 41.6765 %
Root relative squared error             50.6646 %
Total Number of Instances             1593

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.969    0.001    0.987      0.969    0.978      0.976    0.999     0.994     0
                 0.975    0.015    0.883      0.975    0.927      0.919    0.997     0.975     1
                 0.912    0.005    0.954      0.912    0.932      0.925    0.996     0.974     2
                 0.950    0.009    0.921      0.950    0.935      0.928    0.996     0.974     3
                 0.907    0.007    0.936      0.907    0.921      0.913    0.997     0.978     4
                 0.950    0.006    0.950      0.950    0.950      0.944    0.998     0.987     5
                 0.963    0.006    0.945      0.963    0.954      0.949    1.000     0.996     6
                 0.937    0.003    0.974      0.937    0.955      0.950    0.995     0.971     7
                 0.929    0.010    0.911      0.929    0.920      0.911    0.992     0.946     8
                 0.886    0.008    0.927      0.886    0.906      0.896    0.991     0.951     9
Weighted Avg.    0.938    0.007    0.939      0.938    0.938      0.931    0.996     0.975

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 156   0   1   0   1   0   1   0   2   0 |   a = 0
   0 158   0   2   1   1   0   0   0   0 |   b = 1
   0   1 145   0   4   0   0   1   6   2 |   c = 2
   0   2   0 151   0   1   0   2   0   3 |   d = 3
   0   6   1   0 146   2   4   1   0   1 |   e = 4
   0   1   0   2   1 151   3   0   1   0 |   f = 5
   1   1   0   0   2   1 155   0   1   0 |   g = 6
   0   5   0   0   1   0   1 148   1   2 |   h = 7
   0   1   4   2   0   1   0   0 144   3 |   i = 8
   1   4   1   7   0   2   0   0   3 140 |   j = 9
```

5.3.5 ZeroR

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     semeion1-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    1593
Attributes:   257
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         161               10.1067 %
Incorrectly Classified Instances      1432               89.8933 %
Kappa statistic                         -0.0006
Mean absolute error                      0.18
Root mean squared error                  0.3
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             1593

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.099    0.101    0.100      0.099    0.100      -0.001   0.497     0.100     0
                 0.895    0.900    0.101      0.895    0.182      -0.005   0.495     0.101     1
                 0.000    0.000    0.000      0.000    0.000      0.000    0.496     0.099     2
                 0.000    0.000    0.000      0.000    0.000      0.000    0.496     0.099     3
                 0.000    0.000    0.000      0.000    0.000      0.000    0.497     0.100     4
                 0.000    0.000    0.000      0.000    0.000      0.000    0.496     0.099     5
                 0.000    0.000    0.000      0.000    0.000      0.000    0.497     0.100     6
                 0.000    0.000    0.000      0.000    0.000      0.000    0.494     0.098     7
                 0.000    0.000    0.000      0.000    0.000      0.000    0.491     0.095     8
                 0.000    0.000    0.000      0.000    0.000      0.000    0.494     0.098     9
Weighted Avg.    0.101    0.102    0.020      0.101    0.029      -0.001   0.495     0.099

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
  16 145   0   0   0   0   0   0   0   0 |   a = 0
  17 145   0   0   0   0   0   0   0   0 |   b = 1
  16 143   0   0   0   0   0   0   0   0 |   c = 2
  16 143   0   0   0   0   0   0   0   0 |   d = 3
  16 145   0   0   0   0   0   0   0   0 |   e = 4
  16 143   0   0   0   0   0   0   0   0 |   f = 5
  16 145   0   0   0   0   0   0   0   0 |   g = 6
  16 142   0   0   0   0   0   0   0   0 |   h = 7
  15 140   0   0   0   0   0   0   0   0 |   i = 8
  16 142   0   0   0   0   0   0   0   0 |   j = 9
```

5.4 Comparison and interpretation of Results

On comparing these algorithms on the dataset we found that 'Random Forest' performed very well (93.60 percent accuracy with 1.17 as std. deviation) followed by Naïve Bayes (85.32 percent accuracy with 1.81 as std. deviation). Here's the analysis why we think the algorithms gave above results:

1. **Random Forest:** This algorithm is resilient to the noise as it uses maximum votes and bagging techniques. It is well suited for multi-class classification problem. Because of how they are constructed (using bagging or boosting) these algorithms handle very well high dimensional spaces as well as large number of training examples. The dataset we have has 256 with more than 1500 records.

2. **Naïve Bayes** – Primary disadvantage of this algorithm is that they consider features to be independent. In this data, the features are not independent as the near-by pixel are related to each other. This independence assumption seems to be causing dip in performance of this algorithm.

3. **Logistic regression** – This algorithm is believed to expect linear features or even features to interact linearly. It is well suited for binary classification. However, it is not ideal for binary/categorical features. The dataset we have is having multiple class and also all the features (256) are binary. Due to this the Logistic regression is not performing well on our dataset.

4. **Decision Tree** – This algorithm is less tolerant to noise and tends to over-fit the data. In the digits data we might have less organized data for each character.

6 Conclusion

In this mini-project, we selected 3 distinct datasets - German Credit Worthiness and Adult Census data – were **Binary Classification** problems, where as Digit Recognition was a **multi-class classification** problem. We had an opportunity to compare and explore – why different algorithms came out on top – in each of these classification problems. We shared our own speculations about the results (above).

German Credit Worthiness: Top performer was **Naive Bayes**

Adult Census data: Top performer was **Decision Tree (J48)**

Digit Recognition: Top performer was **Random Forests**

The comparison helped us better understand each of the algorithms. The WEKA tools for Preprocessing, Exploring and Experimenting gave us good sense of which algorithms to try out and when to anticipate good results – on the datasets that we will encounter in the future.

<---------- End Of Document ---------->