

GP Earnings and Expenses Report: Median GP Earnings Model

Team Members:

Dev Makwana, Anirudh Chintaluri

10/20/2024

Table of Contents

<u>Part 1 – Statement/Project Goal</u>	<u>3</u>
<u>Part 2 – Description of Dataset</u>	<u>3</u>
<u>Part 3 – Pre-Processing</u>	<u>7</u>
<u>Part 4 – Attribute Selection Algorithms & Model Classifiers Used</u>	<u>8</u>
<u>Part 5 – Results and Analysis</u>	<u>13</u>
<u>Part 6 – Conclusion/How to Reproduce Our Model</u>	<u>32</u>
<u>Part 7 - Team Members and Tasks Performed</u>	<u>33</u>

1. Statement/Project Goal

The purpose of this project is to make use of machine learning to predict general practitioner (GP) earnings in the United Kingdom. We focus specifically on information based on demographic data and what these GPs specialize in. These demographics include, but are not limited to: GP type, contract type, country of residence within the UK, rurality, region, and working hours.

This project will be useful in real-world applications because it allows for better understanding of the circumstances for GPs from specific demographics. Additionally, this can make way for better healthcare policy decisions from governments and healthcare firms, as well as using it as a more generalized metric for economic forecasting, the job market, and quality of life in these regions.

2. Description of Dataset

There was one dataset used, which was downloaded from the National Health Service England website. The dataset contained financial and general information about general practitioners in the UK.

Before preprocessing, we had 55 attributes, 1 class attribute, and 1406 instances to train and test our model with. The dataset is composed of the following attributes:

Attribute Name	Description and possible values (as necessary). Money is in pounds.
GP Type	Type of practitioner. Either salaried, contracted, or combined.
Contract Type	One of three types of contracts: GMS (General Medical Services), or PMS (Personal Medical Services), or GPMS, which is a combination of the previous two.
Country (within UK)	Either England, Wales, Scotland, or Northern

	Ireland.
Practice Type	Either dispensing, non-dispensing, or all.
Gender	Either male, female, or combined.
Age	Nominal. Values are: All, < 40, 40-50, 50-60, 60+
Rurality	Rural/Urban area.
Region	The area in the United Kingdom of which the region is part of. Possible values are: East of England, London, Midlands, North East and Yorkshire, North West, South East of England, and South West of England.
Practiced Registered Patients	Nominal. Values are: < 5000, 5000-10000, 10000-15000, 15000-20000, 20000+
Weekly Working Hours	Number of hours worked each week
Range of Gross Earnings from Self Employment	Nominal. Separated mostly into ranges of 25000, starting at 125,000. A few have separate ranges, such as 325,000+.
Range of Total Earnings from Self Employment	Nominal. Separated mostly into ranges of 25000, starting at 50,000. A few have separate ranges, such as 350,000+.
Range of Income from Self Employment	Nominal. Separated mostly into ranges of 25000, starting at 50,000. A few have separate ranges, such as 125,000+.
Range of Total Income before Tax	Nominal. Separated mostly into ranges of 25000, starting at 0. A few have separate ranges, such as 0-50000 or 100000+.
Sample Count	Continuous values from 50 - 16,700.
Estimated Population	Continuous values from 50 - 31,750.
Average SE Gross Earnings	Shareholder equity gross earnings, continuous from 4,600 - 39,700.
Average SE Expenses	Shareholder equity expenses, continuous from 500 - 22,200.
Average SE Income Before Tax	Shareholder equity income before tax,

	continuous from 2,900 - 21,400.
Average EMP Gross Earnings	Employee gross earnings, continuous from 47,700 - 112,000.
Average EMP Expenses	Employee expenses, continuous from 600 - 3,400.
Average EMP Income Before Tax	Employee income before tax, continuous from 46,400 - 109,700.
Average Tot Gross Earnings	Total gross earnings, continuous from 53,700 - 784,000.
Average Tot Expenses	Total expenses, continuous from 1,300 - 620,500.
Average Tot Income Before Tax	Income before tax, continuous from 51,300 - 234,200.
EER	Estimated energy requirement, continuous from 39.5 - 79.1. Units are unclear.
Income Before Tax Standard Error	Continuous from 290 - 15,029.
Median Income Before Tax	Continuous from 45,200 - 244,900.
Average Total Expenses	Continuous from 116,800 - 620,500.
Average Office and General Business	Cost of general business and office expenses. Continuous from 5,300 - 40,300.
Average Premises	Cost of the premises, continuous from 8,600 - 71,000.
Average Employee	Cost of an average employee, continuous from 68,600 - 369,200.
Average Car and Travel	Average cost of car and travel, continuous from 100 - 2,800.
Average Interest	Continuous from 0 - 17,200.
Average Other	Other expenses, including advertisement, entertainment, interest for business where turnover is less than £85,000 and is not reported separately, and expenses for businesses where turnover is low and detailed expenses breakdown is not available. Continuous from 3,200 - 238,600.

Average Net Capital Allowances	Continuous from 0 - 4,500.
%Zero Office and Generate Business	Continuous from 0.1 - 2.6.
%Zero Premises	Continuous from 0.1 - 3.8.
%Zero Employee	Continuous from 0.1 - 7.7.
%Zero Car and Travel	Continuous from 0.6 - 100.
%Zero Interest	Continuous from 2.4 - 78.6.
%Zero Other	Continuous from 0.1 - 2.5.
%Zero Net Capital Allowances	Continuous from 0.4 - 30.7.
Count of GPs	Continuous from 10 - 7,310.
Percentage of GPs	Continuous from 0.7 - 47.7.
Cumulative Percent of GPs	Continuous from 1.1 - 100.2.
GE Median	Continuous from 64,200 - 472,300.
GE Q1	General expenses in the first quarter, continuous from 49,600 - 340,600.
GE Q3	General expenses in the third quarter, continuous from 78,400 - 637,800.
GE D1	General expenses in the first decile, continuous from 34,600 - 259,100.
GE D9	General expenses in the ninth decile, continuous from 103,100 - 836,500.
TE Median	Travel and expenses cost, continuous from 1,600 - 333,100.
IBT Q1	Income before taxes in the first quarter, continuous from 45,800 - 98,200.
IBT Q3	Income before taxes in the third quarter, continuous from 74,200 - 169,400.
IBT D1	Income before taxes in the first decile, continuous from 33,000 - 73,700.
IBT D9	Income before taxes in the ninth decile,

	continuous from 95,600 - 220,600.
--	-----------------------------------

3. Preprocessing

All pre-processing was done on Weka.

3.1 – Remove Instances Missing the Class

Some instances were missing values for the class variable, Median Income Before Tax. These instances were removed, because it is not possible to run supervised training using instances that do not have a label.

3.2 – Removing unnecessary attributes

There was only one attribute that was clearly not relevant to the class variable: Weekly Working Hours. This attribute was nominal with only one distinct value, so it had no effect on the class. For this reason, we removed this attribute from the dataset.

3.3 – Remove Attributes With Too Many Missing Values

Some attributes were missing over 80% of their values. We removed these attributes entirely, since replacing these missing values with the means or modes of their respective attributes may cause large amounts of bias. We chose 80% as an arbitrary cutoff value. Below are the attributes removed.

- Range_of_Gross_Earnings_from_Self_Employment
- Range of Total Earnings from Self Employment
- Range of Income from Self Employment
- Range of Total Income before Tax
- Average SE Gross Earnings
- Average SE Expenses
- Average SE Income Before Tax
- Average EMP Gross Earnings
- Average EMP Expenses
- Average EMP Income Before Tax
- %Zero Office and Generate Business
- %Zero Premises

- %Zero Employee
- %Zero Other
- GE Median
- GE Q1
- GE Q3
- GE Q3
- GE D9
- TE Median
- IBT Q1
- IBT Q3
- IBT D1
- IBT D9

3.4 – Replacing attributes too similar to the class

Three attributes, Average Tot Gross Earnings, Average Tot Income Before Tax, and Income Before Tax Standard Error, were too similar to the class. In order to make a classification model that can accurately predict the median income before tax of GPs, assuming there is no available data for the three mentioned attributes, we removed the attributes of the model.

3.5 – Replacing missing values

Most attributes had missing values or disguised missing values. These disguised missing values were called ‘All,’ appearing in most of the nominal attributes. This value did not make sense in the context of the attributes. For example, ‘All’ was the value with the highest frequency in the attribute for age of the general practitioner, which does not make sense. Because of this, we counted ‘All’ as a missing value. We filled in all missing values using Weka.

3.6 – Normalization

The values in each attribute followed different scales, with some being from 0-100 and others numbering in the hundreds of thousands. To fix this, we used z-score normalization, ensuring that all attributes have equal weightage during training. All of the quantitative attributes were normalized.

3.7 – Split final dataset into training and testing datasets

We did a 80%/20% split for the training and testing datasets, where 20% of the dataset will be used to test the model's accuracy after using the other 80% to train the model. This split results in 856 instances for training and 214 for testing.

4. Attribute Selection Algorithms & Model Classifiers Used

After pre-processing, the dataset had 22 attributes. In order to use a classification model, we converted the class attribute from a numeric data type to nominal by discretizing it in Weka into four bins of equal width.

Class attribute: Median Income Before Tax

Features: GP_Type, Contract_Type, Country, Practice_Type, Gender, Age, Sample Count, Estimated Population, Average Tot Expenses, EER, Average Total Expenses, Average Office and General Business, Average Premises, Average Employee, Average Car and Travel, Average Interest, Average Other, Average Net Capital Allowances, %Zero Employee, %Zero Car and Travel, %Zero Interest, %Zero Net Capital Allowances

4.1 – Attribute Selection Algorithms

We used Weka to run all of the attribute selection algorithms.

4.1.1 – CorrelationAttributeEval

One method to remove unnecessary attributes is to find how much they affect the class attribute. This is commonly done by finding its Pearson correlation coefficient, given by the following function, where the inputs x and y are the attribute and class:

$$corr(x, y) = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

The CorrelationAttributeEval algorithm uses this formula to find the correlation coefficient for each attribute and the class, ranking them from highest to lowest correlation. We used an arbitrary cutoff value of 0.05 to remove all attributes with correlation coefficients below the cutoff value. The attributes retained with this algorithm are below:

```

=== Attribute Selection on all input data ===

Search Method:
    Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 23 'Median Income Before Tax'):
    Correlation Ranking Filter
Ranked attributes:
0.5473    9 'Average Tot Expenses'
0.4567    1 GP_Type
0.1494    4 Practice_Type
0.1069    5 Gender
0.0971   17 'Average Other'
0.0836   11 'Average Total Expenses'
0.0813    6 Age
0.0785   10 EER
0.069     19 '%Zero Employee'
0.0665   14 'Average Employee'
0.0656   12 'Average Office and General Business'
0.0655    7 'Sample Count'
0.0654   16 'Average Interest'
0.06     18 Average Net Capital Allowances'
0.0539   20 '%Zero Car and Travel'
0.0531    3 Country
0.0386   13 'Average Premises'
0.0347   21 '%Zero Interest'
0.0346    8 'Estimated Population'
0.0169   22 '%Zero Net Capital Allowances'
0.0139   15 'Average Car and Travel'
0.0106    2 Contract_Type

Selected attributes: 9,1,4,5,17,11,6,10,19,14,12,7,16,18,20,3,13,21,8,22,15,2 : 22

```

4.1.2 – OneRAttributeEval

The OneRAttributeEval algorithm uses the following pseudocode to find a single rule to predict the class using a single attribute with the lowest error rate;

For each attribute

For each unique value of the attribute

count the frequency of each class value

find the most frequent class value

make rule where the most frequent class value is assigned to this value of the attribute

Calculate the error rate of each rule for this attribute

Choose the rule with the lowest error rate

The retained attributes with this data selection algorithm are below. These attributes were chosen by those that had a score greater than 68.0.

Attribute Evaluator (supervised, Class (nominal): 23 'Median Income Before Tax'):
OneR feature evaluator.

Using 10 fold cross validation for evaluating attributes.
Minimum bucket size for OneR: 6

Ranked attributes:

86.63551	9	'Average Tot Expenses'
86.26168	1	GP_Type
85.98131	10	EER
78.03738	13	'Average Premises'
77.85047	17	'Average Other'
77.75701	11	'Average Total Expenses'
77.47664	12	'Average Office and General Business'
77.38318	15	'Average Car and Travel'
77.38318	18	Average Net Capital Allowances'
77.00935	16	'Average Interest'
76.82243	14	'Average Employee'
68.03738	20	'%Zero Car and Travel'
68.03738	4	Practice_Type
68.03738	2	Contract_Type
68.03738	3	Country
68.03738	5	Gender
68.03738	6	Age
67.94393	22	'%Zero Net Capital Allowances'
67.94393	21	'%Zero Interest'
67.85047	19	'%Zero Employee'
67.19626	7	'Sample Count'
66.63551	8	'Estimated Population'

Selected attributes: 9,1,10,13,17,11,12,15,18,16,14,20,4,2,3,5,6,22,21,19,7,8 : 22

4.1.3 – InfoGainAttributeEval

This algorithm uses the following formulas to calculate Gain(A) for each attribute A:

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

$$\text{Info}(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

where p_i is the probability that a tuple in D belongs to class C_i , estimated by $\frac{|C_{i,D}|}{|D|}$ and m is the number of classes. Info is the expected information needed to classify a tuple in D. After calculating the gain for each attribute, the attribute with the highest information gain is selected as the best attribute.

The retained attributes with this algorithm are below. These attributes were selected because they had an Information Gain that was above 0.05.

```
Attribute Evaluator (supervised, Class (nominal): 23 'Median Income Before Tax'):  
Information Gain Ranking Filter
```

```
Ranked attributes:
```

```
0.58185    9 'Average Tot Expenses'  
0.5561    10 EER  
0.48809    1 GP_Type  
0.47333    14 'Average Employee'  
0.43973    16 'Average Interest'  
0.42296    11 'Average Total Expenses'  
0.41858    13 'Average Premises'  
0.40513    18 Average Net Capital Allowances'  
0.40381    15 'Average Car and Travel'  
0.38211    12 'Average Office and General Business'  
0.37884    17 'Average Other'  
0.26645    21 '%Zero Interest'  
0.20107    20 '%Zero Car and Travel'  
0.17778    22 %Zero Net Capital Allowances'  
0.11987    19 '%Zero Employee'  
0.096      4 Practice_Type  
0.03429    3 Country  
0.03125    5 Gender  
0.01593    7 'Sample Count'  
0.0132     6 Age  
0.00278    2 Contract_Type  
0          8 'Estimated Population'
```

```
Selected attributes: 9,10,1,14,16,11,13,18,15,12,17,21,20,22,19,4,3,5,7,6,2,8 : 22
```

4.1.4 – CfsSubsetEval

This algorithm evaluates the worth of a subset of attributes. We used the search method GreedyStepwise to find the best subset of attributes. Below are the chosen attributes:

```
Selected attributes: 1,5,9 : 3  
GP_Type  
Gender  
'Average Tot Expenses'
```

4.1.5 – Custom Hand-picked

Based on the attributes selected using the previous attribute selection algorithms, we chose to keep the dataset as it is — meaning that all 22 attributes will remain on this dataset as a means of a control dataset.

4.2 – Classifier Models

4.2.1 – bayes.NaiveBayes

This classifier calculates the following probabilities;

D: Training set of tuples and their respective labels

X: A single tuple with n attributes, with x_i representing the value of the attribute A_i

m classes represented by C_1, C_2, \dots, C_m

$$P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)}$$

where $P(C_1) = (\text{\# yes tuples}) / (\text{total \# tuples})$ and $P(C_2) = (\text{\# no tuples}) / (\text{total \# tuples})$.

Assuming that no attributes depend on any other attributes:

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)$$

4.2.2 – trees.RandomForest

This voting-based classifier creates a forest of random trees. It consists of many separate decision trees that determine a class prediction, where the class with the highest number of trees for each tuple becomes the model's prediction. All trees are equally weighted.

4.2.3 – rules.OneR

This classifier works the same as the OneR attribute selection algorithm in 4.1.2.

4.2.4 – functions.MultilayerPerceptron

This classifier builds and trains a multilayer perceptron using backpropagation to predict the class for each tuple. We used the default settings for the MLP, shown below.

GUI	False
autoBuild	True
batchSize	100
debug	False
decay	False
doNotCheckCapabilities	False
hiddenLayers	a
learningRate	0.3
momentum	0.2
nominalToBinaryFilter	True
normalizeAttributes	True
normalizeNumericClass	True
numDecimalPlaces	2
reset	True
resume	False
seed	0
trainingTime	500
validationSetSize	0
validationThreshold	20

5. Results and Analysis

5.1 – Results

All classification algorithms were run using cross-validation with 10 folds and with the supplied testing set option.

5.1.1 – Results using cross-validation

Correlation with Naive Bayes:

```

Time taken to build model: 0 seconds

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

Correctly Classified Instances      621           72.5467 %
Incorrectly Classified Instances    235           27.4533 %
Kappa statistic                    0.5034
Mean absolute error                0.1385
Root mean squared error            0.3625
Relative absolute error            60.3003 %
Root relative squared error        107.1536 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.913   0.216   0.622    0.913   0.740     0.636   0.923    0.895   '(-inf-95125]'
          0.653   0.120   0.920    0.653   0.764     0.497   0.862    0.899   '(95125-145050]'
          0.625   0.084   0.225    0.625   0.331     0.336   0.921    0.485   '(145050-194975]'
          1.000   0.000   1.000    1.000   1.000     1.000   1.000    1.000   '(194975-inf)'
Weighted Avg.   0.725   0.146   0.811    0.725   0.741     0.531   0.882    0.882

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
219  21   0   0 |  a = '(-inf-95125]'
133 380  69   0 |  b = '(95125-145050]'
  0  12  20   0 |  c = '(145050-194975]'
  0   0   0   2 |  d = '(194975-inf)'

```

Correlation with Random Forest:

```

Time taken to build model: 0.07 seconds

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

Correctly Classified Instances      814           95.0935 %
Incorrectly Classified Instances     42            4.9065 %
Kappa statistic                    0.8903
Mean absolute error                0.0486
Root mean squared error            0.1443
Relative absolute error            21.1569 %
Root relative squared error        42.6663 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.933   0.018   0.953    0.933   0.943     0.921   0.991    0.980   '(-inf-95125]'
          0.978   0.106   0.952    0.978   0.964     0.886   0.985    0.993   '(95125-145050]'
          0.594   0.002   0.905    0.594   0.717     0.725   0.992    0.834   '(145050-194975]'
          1.000   0.000   1.000    1.000   1.000     1.000   1.000    1.000   '(194975-inf)'
Weighted Avg.   0.951   0.077   0.950    0.951   0.949     0.890   0.987    0.983

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
224  16   0   0 |  a = '(-inf-95125]'
 11 569   2   0 |  b = '(95125-145050]'
  0  13  19   0 |  c = '(145050-194975]'
  0   0   0   2 |  d = '(194975-inf)'

```

Correlation with OneR:

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances      737           86.0981 %
Incorrectly Classified Instances    119           13.9019 %
Kappa statistic                    0.6729
Mean absolute error                 0.0695
Root mean squared error             0.2636
Relative absolute error             30.2721 %
Root relative squared error         77.9369 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.721   0.029   0.906    0.721   0.803     0.746   0.846    0.731   '(-inf-95125]'
      0.952   0.328   0.860    0.952   0.904     0.674   0.812    0.852   '(95125-145050]'
      0.313   0.013   0.476    0.313   0.377     0.367   0.650    0.175   '(145050-194975]'
      0.000   0.000   ?        0.000   ?         ?       0.500    0.002   '(194975-inf)'
Weighted Avg.  0.861   0.232   ?        0.861   ?         ?       0.814    0.791

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
173  67   0   0 |  a = '(-inf-95125]'
 17 554  11   0 |  b = '(95125-145050]'
  1  21  10   0 |  c = '(145050-194975]'
  0   2   0   0 |  d = '(194975-inf)'
```

Correlation with MLP:


```

Time taken to build model: 1.51 seconds

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

Correctly Classified Instances      792           92.5234 %
Incorrectly Classified Instances    64           7.4766 %
Kappa statistic                    0.831
Mean absolute error                 0.0424
Root mean squared error             0.1596
Relative absolute error             18.466 %
Root relative squared error         47.185 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.900   0.028   0.927     0.900   0.913     0.880   0.986   0.958   '(-inf-95125]'
                0.966   0.157   0.929     0.966   0.947     0.829   0.972   0.982   '(95125-145050]'
                0.438   0.005   0.778     0.438   0.560     0.572   0.916   0.691   '(145050-194975]'
                0.000   0.000   ?         0.000   ?         ?       0.182   0.002   '(194975-inf)'
Weighted Avg.    0.925   0.115   ?         0.925   ?         ?       0.972   0.962

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
216 24  0  0 |  a = '(-inf-95125]'
 17 562 3  0 |  b = '(95125-145050]'
  0 18 14  0 |  c = '(145050-194975]'
  0  1  1  0 |  d = '(194975-inf)'

```

OneR with Naive Bayes:

```

Time taken to build model: 0 seconds

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

Correctly Classified Instances      607           70.9112 %
Incorrectly Classified Instances    249           29.0888 %
Kappa statistic                    0.4875
Mean absolute error                 0.1439
Root mean squared error             0.3709
Relative absolute error             62.6803 %
Root relative squared error         109.6347 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.913   0.226   0.612     0.913   0.732     0.625   0.934   0.904   '(-inf-95125]'
                0.622   0.106   0.926     0.622   0.744     0.483   0.877   0.906   '(95125-145050]'
                0.750   0.098   0.229     0.750   0.350     0.377   0.917   0.499   '(145050-194975]'
                1.000   0.000   1.000     1.000   1.000     1.000   1.000   1.000   '(194975-inf)'
Weighted Avg.    0.709   0.139   0.812     0.709   0.727     0.520   0.895   0.891

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
219 21  0  0 |  a = '(-inf-95125]'
139 362 81  0 |  b = '(95125-145050]'
  0  8 24  0 |  c = '(145050-194975]'
  0  0  0  2 |  d = '(194975-inf)'

```

OneR with Random Forest:

```
Time taken to build model: 0.08 seconds

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

Correctly Classified Instances      801          93.5748 %
Incorrectly Classified Instances    55           6.4252 %
Kappa statistic                    0.8553
Mean absolute error                 0.0535
Root mean squared error             0.1539
Relative absolute error             23.2939 %
Root relative squared error         45.4916 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.908	0.024	0.936	0.908	0.922	0.892	0.991	0.979	'(-inf-95125]'
	0.971	0.139	0.937	0.971	0.954	0.851	0.984	0.992	'(95125-145050]'
	0.500	0.002	0.889	0.500	0.640	0.658	0.986	0.823	'(145050-194975]'
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	'(194975-inf)'
Weighted Avg.	0.936	0.101	0.935	0.936	0.933	0.855	0.986	0.982	

```
=== Confusion Matrix ===

  a  b  c  d  <-- classified as
218 22  0  0 | a = '(-inf-95125]'
```

15	565	2	0		b = '(95125-145050]'
0	16	16	0		c = '(145050-194975]'
0	0	0	2		d = '(194975-inf)'

OneR with OneR:

```

Time taken to build model: 0 seconds

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

Correctly Classified Instances      737           86.0981 %
Incorrectly Classified Instances    119           13.9019 %
Kappa statistic                    0.6729
Mean absolute error                0.0695
Root mean squared error            0.2636
Relative absolute error            30.2721 %
Root relative squared error        77.9369 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.721   0.029   0.906     0.721   0.803     0.746   0.846    0.731    '(-inf-95125]'
                0.952   0.328   0.860     0.952   0.904     0.674   0.812    0.852    '(95125-145050]'
                0.313   0.013   0.476     0.313   0.377     0.367   0.650    0.175    '(145050-194975]'
                0.000   0.000   ?         0.000   ?         ?       0.500    0.002    '(194975-inf)'
Weighted Avg.   0.861   0.232   ?         0.861   ?         ?       0.814    0.791

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
173 67  0  0 |  a = '(-inf-95125]'
 17 554 11  0 |  b = '(95125-145050]'
  1 21 10  0 |  c = '(145050-194975]'
  0  2  0  0 |  d = '(194975-inf)'

```

OneR with MLP:

```

Time taken to build model: 1.78 seconds

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

Correctly Classified Instances      767           89.6028 %
Incorrectly Classified Instances     89           10.3972 %
Kappa statistic                    0.7578
Mean absolute error                0.0525
Root mean squared error            0.1937
Relative absolute error            22.8597 %
Root relative squared error        57.2488 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.838   0.032   0.910     0.838   0.872     0.826   0.976    0.942    '(-inf-95125]'
                0.962   0.245   0.893     0.962   0.926     0.756   0.947    0.958    '(95125-145050]'
                0.188   0.002   0.750     0.188   0.300     0.365   0.895    0.647    '(145050-194975]'
                0.000   0.000   ?         0.000   ?         ?       0.534    0.006    '(194975-inf)'
Weighted Avg.   0.896   0.175   ?         0.896   ?         ?       0.952    0.940

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
201 39  0  0 |  a = '(-inf-95125]'
 20 560  2  0 |  b = '(95125-145050]'
  0 26  6  0 |  c = '(145050-194975]'
  0  2  0  0 |  d = '(194975-inf)'

```

Info Gain with Naive Bayes:

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances      603           70.4439 %
Incorrectly Classified Instances    253           29.5561 %
Kappa statistic                    0.4777
Mean absolute error                 0.1461
Root mean squared error             0.3745
Relative absolute error             63.6433 %
Root relative squared error         110.7033 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.913	0.235	0.602	0.913	0.725	0.615	0.930	0.884	'(-inf-95125]'
	0.619	0.113	0.921	0.619	0.740	0.473	0.875	0.900	'(95125-145050]'
	0.688	0.093	0.222	0.688	0.336	0.352	0.889	0.447	'(145050-194975]'
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	'(194975-inf)'
Weighted Avg.	0.704	0.146	0.805	0.704	0.721	0.510	0.891	0.879	

```
=== Confusion Matrix ===

  a   b   c   d  <-- classified as
219 21   0   0 |  a = '(-inf-95125]'
```

145	360	77	0		b = '(95125-145050]'
0	10	22	0		c = '(145050-194975]'
0	0	0	2		d = '(194975-inf)'

Info Gain with Random Forest:

Time taken to build model: 0.12 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	814	95.0935 %
Incorrectly Classified Instances	42	4.9065 %
Kappa statistic	0.8904	
Mean absolute error	0.0473	
Root mean squared error	0.15	
Relative absolute error	20.6174 %	
Root relative squared error	44.348 %	
Total Number of Instances	856	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.925	0.018	0.953	0.925	0.939	0.915	0.989	0.964	'(-inf-95125]'
	0.978	0.106	0.952	0.978	0.964	0.886	0.981	0.990	'(95125-145050]'
	0.656	0.002	0.913	0.656	0.764	0.767	0.989	0.844	'(145050-194975]'
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	'(194975-inf)'
Weighted Avg.	0.951	0.077	0.951	0.951	0.950	0.890	0.984	0.977	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
222	18	0	0	a = '(-inf-95125]'
11	569	2	0	b = '(95125-145050]'
0	11	21	0	c = '(145050-194975]'
0	0	0	2	d = '(194975-inf)'

Info Gain with OneR:

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	737	86.0981 %
Incorrectly Classified Instances	119	13.9019 %
Kappa statistic	0.6729	
Mean absolute error	0.0695	
Root mean squared error	0.2636	
Relative absolute error	30.2721 %	
Root relative squared error	77.9369 %	
Total Number of Instances	856	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.721	0.029	0.906	0.721	0.803	0.746	0.846	0.731	'(-inf-95125]'
	0.952	0.328	0.860	0.952	0.904	0.674	0.812	0.852	'(95125-145050]'
	0.313	0.013	0.476	0.313	0.377	0.367	0.650	0.175	'(145050-194975]'
	0.000	0.000	?	0.000	?	?	0.500	0.002	'(194975-inf)'
Weighted Avg.	0.861	0.232	?	0.861	?	?	0.814	0.791	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
173	67	0	0	a = '(-inf-95125]'
17	554	11	0	b = '(95125-145050]'
1	21	10	0	c = '(145050-194975]'
0	2	0	0	d = '(194975-inf)'

Info Gain with MLP:

```
Time taken to build model: 0.94 seconds

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

Correctly Classified Instances      807           94.2757 %
Incorrectly Classified Instances    49           5.7243 %
Kappa statistic                    0.8714
Mean absolute error                0.0389
Root mean squared error            0.1555
Relative absolute error            16.9245 %
Root relative squared error        45.9549 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.016	0.957	0.933	0.945	0.924	0.983	0.951	'(-inf-95125]'
	0.974	0.124	0.943	0.974	0.959	0.867	0.961	0.969	'(95125-145050]'
	0.500	0.006	0.762	0.500	0.604	0.606	0.944	0.698	'(145050-194975]'
	0.000	0.000	?	0.000	?	?	0.152	0.002	'(194975-inf)'
Weighted Avg.	0.943	0.089	?	0.943	?	?	0.965	0.951	

```
=== Confusion Matrix ===

  a  b  c  d  <-- classified as
224 16  0  0 |  a = '(-inf-95125]'
```

10	567	5	0		b = '(95125-145050]'
0	16	16	0		c = '(145050-194975]'
0	2	0	0		d = '(194975-inf)'

CfsSubset with Naive Bayes:

```

Time taken to build model: 0 seconds

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

Correctly Classified Instances      753           87.9673 %
Incorrectly Classified Instances    103           12.0327 %
Kappa statistic                    0.7166
Mean absolute error                 0.0858
Root mean squared error             0.2224
Relative absolute error             37.3531 %
Root relative squared error         65.745 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0.738   0.018   0.941    0.738   0.827     0.781   0.953    0.917    '(-inf-95125]'
          0.967   0.303   0.872    0.967   0.917     0.720   0.898    0.926    '(95125-145050]'
          0.406   0.005   0.765    0.406   0.531     0.546   0.929    0.584    '(145050-194975]'
          0.000   0.006   0.000    0.000   0.000    -0.004   0.991    0.156    '(194975-inf)'
Weighted Avg.    0.880   0.211   0.885    0.880   0.875     0.729   0.915    0.909

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
177  62   0   1 |  a = '(-inf-95125]'
 11 563   4   4 |  b = '(95125-145050]'
  0  19  13   0 |  c = '(145050-194975]'
  0   2   0   0 |  d = '(194975-inf)'

```

CfsSubset with Random Forest:

```

Time taken to build model: 0.03 seconds

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

Correctly Classified Instances      760           88.785 %
Incorrectly Classified Instances    96           11.215 %
Kappa statistic                    0.7591
Mean absolute error                 0.0668
Root mean squared error             0.2108
Relative absolute error             29.1047 %
Root relative squared error        62.312 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.896   0.065   0.843     0.896   0.869     0.816   0.960    0.917    '(-inf-95125]'
                0.904   0.146   0.929     0.904   0.916     0.747   0.932    0.953    '(95125-145050]'
                0.531   0.018   0.531     0.531   0.531     0.513   0.882    0.460    '(145050-194975]'
                1.000   0.001   0.667     1.000   0.800     0.816   0.999    0.583    '(194975-inf)'
Weighted Avg.    0.888   0.118   0.890     0.888   0.888     0.758   0.939    0.923

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
215  25   0   0 |  a = '(-inf-95125]'
 40 526  15   1 |  b = '(95125-145050]'
  0  15  17   0 |  c = '(145050-194975]'
  0   0   0   2 |  d = '(194975-inf)'

```

CfsSubset with OneR:

```

Time taken to build model: 0 seconds

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

Correctly Classified Instances      740           86.4486 %
Incorrectly Classified Instances    116           13.5514 %
Kappa statistic                    0.683
Mean absolute error                 0.0678
Root mean squared error             0.2603
Relative absolute error             29.5089 %
Root relative squared error        76.9482 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.729   0.029   0.907     0.729   0.808     0.752   0.850    0.737    '(-inf-95125]'
                0.950   0.318   0.864     0.950   0.905     0.679   0.816    0.855    '(95125-145050]'
                0.375   0.013   0.522     0.375   0.436     0.424   0.681    0.219    '(145050-194975]'
                0.000   0.000   ?          0.000   ?          ?       0.500    0.002    '(194975-inf)'
Weighted Avg.    0.864   0.225   ?          0.864   ?          ?       0.820    0.796

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
175  65   0   0 |  a = '(-inf-95125]'
 18 553  11   0 |  b = '(95125-145050]'
  0  20  12   0 |  c = '(145050-194975]'
  0   2   0   0 |  d = '(194975-inf)'

```


CfsSubset with MLP:

```
Time taken to build model: 0.21 seconds

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

Correctly Classified Instances      757           88.4346 %
Incorrectly Classified Instances     99           11.5654 %
Kappa statistic                     0.7248
Mean absolute error                  0.0908
Root mean squared error              0.2153
Relative absolute error              39.561 %
Root relative squared error          63.6582 %
Total Number of Instances           856

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.742	0.018	0.942	0.742	0.830	0.784	0.953	0.917	'(-inf-95125]'
	0.974	0.307	0.871	0.974	0.920	0.730	0.908	0.930	'(95125-145050]'
	0.375	0.005	0.750	0.375	0.500	0.518	0.932	0.561	'(145050-194975]'
	0.000	0.000	?	0.000	?	?	0.181	0.002	'(194975-inf)'
Weighted Avg.	0.884	0.214	?	0.884	?	?	0.920	0.910	

```
=== Confusion Matrix ===

  a  b  c  d  <-- classified as
178 62  0  0 |  a = '(-inf-95125]'
```

11	567	4	0		b = '(95125-145050]'
0	20	12	0		c = '(145050-194975]'
0	2	0	0		d = '(194975-inf)'

Custom with Naive Bayes:

```

Time taken to build model: 0 seconds

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

Correctly Classified Instances      601          70.2103 %
Incorrectly Classified Instances    255          29.7897 %
Kappa statistic                    0.4727
Mean absolute error                0.1491
Root mean squared error            0.3789
Relative absolute error            64.953 %
Root relative squared error        111.9989 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.908    0.234    0.602    0.908    0.724      0.613    0.930    0.893    '(-inf-95125]'
      0.619    0.120    0.916    0.619    0.738      0.466    0.873    0.902    '(95125-145050]'
      0.656    0.095    0.212    0.656    0.321      0.333    0.894    0.460    '(145050-194975]'
      1.000    0.000    1.000    1.000    1.000      1.000    1.000    1.000    '(194975-inf)'
Weighted Avg.    0.702    0.151    0.802    0.702    0.719      0.504    0.890    0.883

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
218  22   0   0 |  a = '(-inf-95125]'
144 360  78   0 |  b = '(95125-145050]'
  0  11  21   0 |  c = '(145050-194975]'
  0   0   0   2 |  d = '(194975-inf)'

```

Custom with Random Forest:

```

Time taken to build model: 0.07 seconds

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

Correctly Classified Instances      809          94.5093 %
Incorrectly Classified Instances     47           5.4907 %
Kappa statistic                    0.8769
Mean absolute error                0.055
Root mean squared error            0.1503
Relative absolute error            23.9702 %
Root relative squared error        44.4309 %
Total Number of Instances         856

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.929    0.021    0.945    0.929    0.937      0.913    0.992    0.982    '(-inf-95125]'
      0.974    0.117    0.947    0.974    0.960      0.873    0.985    0.992    '(95125-145050]'
      0.531    0.002    0.895    0.531    0.667      0.681    0.988    0.834    '(145050-194975]'
      1.000    0.000    1.000    1.000    1.000      1.000    1.000    1.000    '(194975-inf)'
Weighted Avg.    0.945    0.085    0.944    0.945    0.943      0.877    0.987    0.984

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
223  17   0   0 |  a = '(-inf-95125]'
 13 567   2   0 |  b = '(95125-145050]'
  0  15  17   0 |  c = '(145050-194975]'
  0   0   0   2 |  d = '(194975-inf)'

```

Custom with OneR:

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances      737           86.0981 %
Incorrectly Classified Instances    119           13.9019 %
Kappa statistic                    0.6729
Mean absolute error                 0.0695
Root mean squared error             0.2636
Relative absolute error             30.2721 %
Root relative squared error         77.9369 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.721	0.029	0.906	0.721	0.803	0.746	0.846	0.731	'(-inf-95125]'
	0.952	0.328	0.860	0.952	0.904	0.674	0.812	0.852	'(95125-145050]'
	0.313	0.013	0.476	0.313	0.377	0.367	0.650	0.175	'(145050-194975]'
	0.000	0.000	?	0.000	?	?	0.500	0.002	'(194975-inf)'
Weighted Avg.	0.861	0.232	?	0.861	?	?	0.814	0.791	

```
=== Confusion Matrix ===

  a   b   c   d  <-- classified as
173  67   0   0 |  a = '(-inf-95125]'
```

17	554	11	0		b = '(95125-145050]'
1	21	10	0		c = '(145050-194975]'
0	2	0	0		d = '(194975-inf)'

Custom with MLP:

```

Time taken to build model: 2.48 seconds

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

Correctly Classified Instances      771           90.0701 %
Incorrectly Classified Instances    85           9.9299 %
Kappa statistic                    0.7701
Mean absolute error                 0.0503
Root mean squared error             0.191
Relative absolute error             21.8936 %
Root relative squared error         56.4518 %
Total Number of Instances          856

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.875   0.037   0.901     0.875   0.888     0.845   0.976   0.957   '(-inf-95125]'
                0.959   0.223   0.901     0.959   0.929     0.767   0.939   0.947   '(95125-145050]'
                0.094   0.001   0.750     0.094   0.167     0.257   0.910   0.596   '(145050-194975]'
                0.000   0.000   ?         0.000   ?         ?       0.060   0.002   '(194975-inf)'
Weighted Avg.   0.901   0.162   ?         0.901   ?         ?       0.947   0.934

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
210  30   0   0 |  a = '(-inf-95125]'
 23 558   1   0 |  b = '(95125-145050]'
  0  29   3   0 |  c = '(145050-194975]'
  0   2   0   0 |  d = '(194975-inf)'

```

5.1.2 – Results using test set

Correlation with Naive Bayes:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances      146          70.8738 %
Incorrectly Classified Instances    60           29.1262 %
Kappa statistic                    0.4537
Mean absolute error                 0.1521
Root mean squared error             0.3798
Relative absolute error             69.4694 %
Root relative squared error         117.8476 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.867   0.205   0.634    0.867   0.732     0.614    0.904    0.879    '(-inf-95125]'
      0.644   0.133   0.922    0.644   0.758     0.464    0.832    0.881    '(95125-145050]'
      ?       0.107   0.000    ?       ?         ?        ?        ?        '(145050-194975]'
      ?       0.000   ?        ?       ?         ?        ?        ?        '(194975-inf)'
Weighted Avg.  0.709   0.154   0.838    0.709   0.751     0.508    0.853    0.881

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
52  8  0  0 | a = '(-inf-95125]'
30 94 22  0 | b = '(95125-145050]'
 0  0  0  0 | c = '(145050-194975]'
 0  0  0  0 | d = '(194975-inf)'

```

Correlation with Random Forest:

```

Time taken to build model: 0.07 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      201          97.5728 %
Incorrectly Classified Instances     5           2.4272 %
Kappa statistic                     0.9413
Mean absolute error                  0.0377
Root mean squared error              0.1145
Relative absolute error              17.1982 %
Root relative squared error          35.5224 %
Total Number of Instances           206

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.950    0.000    1.000     0.950   0.974     0.965    0.998     0.996    '(-inf-95125]'
          0.986    0.050    0.980     0.986   0.983     0.941    0.995     0.998    '(95125-145050]'
          ?         0.010    0.000     ?       ?         ?       ?       ?       '(145050-194975]'
          ?         0.000    ?         ?       ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.  0.976    0.035    0.986     0.976   0.980     0.948    0.996     0.997

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
57  3  0  0 |  a = '(-inf-95125]'
 0 144  2  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

Correlation with OneR:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      182          88.3495 %
Incorrectly Classified Instances    24          11.6505 %
Kappa statistic                    0.7161
Mean absolute error                 0.0583
Root mean squared error            0.2414
Relative absolute error             26.6041 %
Root relative squared error        74.8939 %
Total Number of Instances         206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC     ROC Area  PRC Area  Class
      0.767   0.041   0.885    0.767   0.821     0.759   0.863    0.746    '(-inf-95125]'
      0.932   0.233   0.907    0.932   0.919     0.713   0.849    0.893    '(95125-145050]'
      ?       0.019   0.000    ?       ?         ?       ?       ?       '(145050-194975]'
      ?       0.000   ?        ?       ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.  0.883   0.177   0.900    0.883   0.891     0.726   0.853    0.850

=== Confusion Matrix ===

 a  b  c  d  <-- classified as
46 14  0  0 |  a = '(-inf-95125]'
 6 136 4  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

Correlation with MLP:

```

Time taken to build model: 1.49 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      196          95.1456 %
Incorrectly Classified Instances    10           4.8544 %
Kappa statistic                    0.8789
Mean absolute error                 0.0294
Root mean squared error             0.1296
Relative absolute error             13.4208 %
Root relative squared error         40.2052 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.867    0.014    0.963    0.867    0.912      0.881    0.980    0.976    '(-inf-95125]'
      0.986    0.133    0.947    0.986    0.966      0.881    0.976    0.986    '(95125-145050]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(194975-inf)'
Weighted Avg.  0.951    0.098    0.952    0.951    0.951      0.881    0.977    0.983

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
52   8   0   0 |  a = '(-inf-95125]'
 2 144   0   0 |  b = '(95125-145050]'
 0   0   0   0 |  c = '(145050-194975]'
 0   0   0   0 |  d = '(194975-inf)'

```

OneR with Naive Bayes:


```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      138           66.9903 %
Incorrectly Classified Instances    68           33.0097 %
Kappa statistic                    0.4167
Mean absolute error                 0.1673
Root mean squared error            0.3974
Relative absolute error             76.4088 %
Root relative squared error        123.3023 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.900    0.233    0.614    0.900    0.730     0.613    0.926    0.897    '(-inf-95125]'
      0.575    0.100    0.933    0.575    0.712     0.435    0.857    0.893    '(95125-145050]'
      ?        0.136    0.000    ?        ?         ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?         ?        ?        ?        '(194975-inf)'
Weighted Avg.    0.670    0.139    0.840    0.670    0.717     0.487    0.877    0.894

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
54  6  0  0 | a = '(-inf-95125]'
34 84 28  0 | b = '(95125-145050]'
 0  0  0  0 | c = '(145050-194975]'
 0  0  0  0 | d = '(194975-inf)'

```

OneR with Random Forest:

```

Time taken to build model: 0.07 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      200          97.0874 %
Incorrectly Classified Instances     6           2.9126 %
Kappa statistic                     0.9297
Mean absolute error                  0.0393
Root mean squared error              0.1169
Relative absolute error              17.9367 %
Root relative squared error          36.266 %
Total Number of Instances           206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.950    0.014    0.966    0.950    0.958      0.941    0.997    0.994    '(-inf-95125]'
      0.979    0.050    0.979    0.979    0.979      0.929    0.995    0.998    '(95125-145050]'
      ?        0.005    0.000    ?        ?          ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(194975-inf)'
Weighted Avg.  0.971    0.039    0.976    0.971    0.973      0.933    0.996    0.997

=== Confusion Matrix ===

  a   b   c   d   <-- classified as
57   3   0   0 |   a = '(-inf-95125]'
 2 143   1   0 |   b = '(95125-145050]'
 0   0   0   0 |   c = '(145050-194975]'
 0   0   0   0 |   d = '(194975-inf)'

```

OneR with OneR:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      182           88.3495 %
Incorrectly Classified Instances    24           11.6505 %
Kappa statistic                    0.7161
Mean absolute error                 0.0583
Root mean squared error            0.2414
Relative absolute error            26.6041 %
Root relative squared error        74.8939 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.767    0.041    0.885    0.767    0.821      0.759    0.863    0.746    '(-inf-95125]'
      0.932    0.233    0.907    0.932    0.919      0.713    0.849    0.893    '(95125-145050]'
      ?        0.019    0.000    ?        ?          ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(194975-inf)'
Weighted Avg.    0.883    0.177    0.900    0.883    0.891      0.726    0.853    0.850

=== Confusion Matrix ===

  a   b   c   d   <-- classified as
46  14   0   0 |   a = '(-inf-95125]'
 6 136   4   0 |   b = '(95125-145050]'
 0   0   0   0 |   c = '(145050-194975]'
 0   0   0   0 |   d = '(194975-inf)'

```

OneR with MLP:

```

Time taken to build model: 1.8 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      200          97.0874 %
Incorrectly Classified Instances     6           2.9126 %
Kappa statistic                     0.9295
Mean absolute error                  0.0293
Root mean squared error              0.1289
Relative absolute error              13.3882 %
Root relative squared error          39.9958 %
Total Number of Instances           206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.950    0.021    0.950    0.950    0.950      0.929    0.974    0.902    '(-inf-95125]'
      0.979    0.050    0.979    0.979    0.979      0.929    0.976    0.982    '(95125-145050]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(194975-inf)'
Weighted Avg.  0.971    0.041    0.971    0.971    0.971      0.929    0.975    0.959

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
57   3   0   0 |  a = '(-inf-95125]'
 3 143   0   0 |  b = '(95125-145050]'
 0   0   0   0 |  c = '(145050-194975]'
 0   0   0   0 |  d = '(194975-inf)'

```

Info Gain with Naive Bayes:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      139          67.4757 %
Incorrectly Classified Instances    67          32.5243 %
Kappa statistic                    0.4203
Mean absolute error                 0.1663
Root mean squared error             0.3975
Relative absolute error             75.9679 %
Root relative squared error         123.3425 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.900    0.240    0.607    0.900    0.725     0.606    0.918    0.858    '(-inf-95125]'
      0.582    0.100    0.934    0.582    0.717     0.441    0.853    0.903    '(95125-145050]'
      ?        0.126    0.000    ?        ?         ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?         ?        ?        ?        '(194975-inf)'
Weighted Avg.    0.675    0.141    0.839    0.675    0.719     0.489    0.872    0.890

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
54  6  0  0 | a = '(-inf-95125]'
35 85 26  0 | b = '(95125-145050]'
 0  0  0  0 | c = '(145050-194975]'
 0  0  0  0 | d = '(194975-inf)'

```

Info Gain with Random Forest:

```

Time taken to build model: 0.09 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      199          96.6019 %
Incorrectly Classified Instances     7           3.3981 %
Kappa statistic                     0.9187
Mean absolute error                  0.0371
Root mean squared error              0.126
Relative absolute error              16.9393 %
Root relative squared error          39.1009 %
Total Number of Instances           206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.950    0.014    0.966    0.950    0.958      0.941    0.992    0.985    '(-inf-95125]'
      0.973    0.050    0.979    0.973    0.976      0.918    0.984    0.991    '(95125-145050]'
      ?        0.010    0.000    ?        ?          ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?          ?        ?        ?        '(194975-inf)'
Weighted Avg.  0.966    0.039    0.975    0.966    0.971      0.925    0.986    0.989

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
57  3  0  0 |  a = '(-inf-95125]'
 2 142  2  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

Info Gain with OneR:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      182          88.3495 %
Incorrectly Classified Instances    24          11.6505 %
Kappa statistic                    0.7161
Mean absolute error                 0.0583
Root mean squared error             0.2414
Relative absolute error             26.6041 %
Root relative squared error         74.8939 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.767    0.041    0.885    0.767    0.821     0.759    0.863    0.746    '(-inf-95125]'
      0.932    0.233    0.907    0.932    0.919     0.713    0.849    0.893    '(95125-145050]'
      ?        0.019    0.000    ?        ?         ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?         ?        ?        ?        '(194975-inf)'
Weighted Avg.  0.883    0.177    0.900    0.883    0.891     0.726    0.853    0.850

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
46  14   0   0 |  a = '(-inf-95125]'
 6 136   4   0 |  b = '(95125-145050]'
 0   0   0   0 |  c = '(145050-194975]'
 0   0   0   0 |  d = '(194975-inf)'

```

Info Gain with MLP:

```

Time taken to build model: 0.95 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      199          96.6019 %
Incorrectly Classified Instances     7           3.3981 %
Kappa statistic                    0.9173
Mean absolute error                 0.0252
Root mean squared error             0.1253
Relative absolute error             11.526 %
Root relative squared error         38.8682 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.933    0.021    0.949    0.933    0.941     0.917    0.971    0.954    '(-inf-95125]'
      0.979    0.067    0.973    0.979    0.976     0.917    0.968    0.983    '(95125-145050]'
      ?        0.000    ?        ?        ?         ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?         ?        ?        ?        '(194975-inf)'
Weighted Avg.    0.966    0.053    0.966    0.966    0.966     0.917    0.969    0.975

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
56  4  0  0 |  a = '(-inf-95125]'
 3 143  0  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

CfsSubset with Naive Bayes:


```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      189          91.7476 %
Incorrectly Classified Instances    17           8.2524 %
Kappa statistic                    0.7887
Mean absolute error                 0.0815
Root mean squared error             0.2099
Relative absolute error             37.2217 %
Root relative squared error         65.12 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
              0.767   0.021   0.939     0.767   0.844     0.796   0.952    0.916    '(-inf-95125]'
              0.979   0.233   0.911     0.979   0.944     0.796   0.905    0.945    '(95125-145050]'
              ?       0.000   ?         ?         ?         ?       ?       ?       '(145050-194975]'
              ?       0.000   ?         ?         ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.   0.917   0.171   0.919     0.917   0.915     0.796   0.918    0.937

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
46 14  0  0 |  a = '(-inf-95125]'
 3 143  0  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

CfsSubset with Random Forest:

```

Time taken to build model: 0.03 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      194          94.1748 %
Incorrectly Classified Instances    12           5.8252 %
Kappa statistic                    0.8635
Mean absolute error                 0.04
Root mean squared error             0.154
Relative absolute error             18.2462 %
Root relative squared error         47.7727 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.933    0.027    0.933    0.933    0.933     0.906    0.983    0.972    '(-inf-95125]'
      0.945    0.067    0.972    0.945    0.958     0.863    0.977    0.985    '(95125-145050]'
      ?        0.019    0.000    ?        ?         ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?         ?        ?        ?        '(194975-inf)'
Weighted Avg.    0.942    0.055    0.961    0.942    0.951     0.875    0.979    0.981

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
56   4   0   0 |  a = '(-inf-95125]'
 4 138   4   0 |  b = '(95125-145050]'
 0   0   0   0 |  c = '(145050-194975]'
 0   0   0   0 |  d = '(194975-inf)'

```

CfsSubset with OneR:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      182           88.3495 %
Incorrectly Classified Instances    24           11.6505 %
Kappa statistic                     0.7161
Mean absolute error                 0.0583
Root mean squared error            0.2414
Relative absolute error             26.6041 %
Root relative squared error        74.8939 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.767   0.041   0.885    0.767   0.821     0.759   0.863    0.746    '(-inf-95125]'
      0.932   0.233   0.907    0.932   0.919     0.713   0.849    0.893    '(95125-145050]'
      ?       0.019   0.000    ?        ?         ?       ?       ?       '(145050-194975]'
      ?       0.000    ?        ?        ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.  0.883   0.177   0.900    0.883   0.891     0.726   0.853    0.850

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
46 14  0  0 |  a = '(-inf-95125]'
 6 136 4  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

CfsSubset with MLP:

```

Time taken to build model: 0.21 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      190           92.233 %
Incorrectly Classified Instances    16           7.767 %
Kappa statistic                    0.8001
Mean absolute error                 0.0736
Root mean squared error             0.1848
Relative absolute error             33.6046 %
Root relative squared error         57.3329 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0.767   0.014   0.958     0.767   0.852     0.809   0.957    0.924    '(-inf-95125]'
          0.986   0.233   0.911     0.986   0.947     0.809   0.940    0.969    '(95125-145050]'
          ?       0.000   ?         ?         ?         ?       ?       ?       '(145050-194975]'
          ?       0.000   ?         ?         ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.  0.922   0.169   0.925     0.922   0.920     0.809   0.945    0.956

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
46  14   0   0 |  a = '(-inf-95125]'
 2 144   0   0 |  b = '(95125-145050]'
 0   0   0   0 |  c = '(145050-194975]'
 0   0   0   0 |  d = '(194975-inf)'

```

Custom with Naive Bayes:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      141          68.4466 %
Incorrectly Classified Instances    65          31.5534 %
Kappa statistic                    0.4279
Mean absolute error                 0.1632
Root mean squared error             0.3954
Relative absolute error             74.5473 %
Root relative squared error        122.7067 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.883    0.226    0.616    0.883    0.726     0.606    0.915    0.884    '(-inf-95125]'
      0.603    0.117    0.926    0.603    0.730     0.443    0.845    0.892    '(95125-145050]'
      ?        0.121    0.000    ?        ?        ?        ?        ?        '(145050-194975]'
      ?        0.000    ?        ?        ?        ?        ?        ?        '(194975-inf)'
Weighted Avg.    0.684    0.149    0.836    0.684    0.729     0.490    0.866    0.889

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
53  7  0  0 | a = '(-inf-95125]'
33 88 25  0 | b = '(95125-145050]'
 0  0  0  0 | c = '(145050-194975]'
 0  0  0  0 | d = '(194975-inf)'

```

Custom with Random Forest:

```

Time taken to build model: 0.08 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      196          95.1456 %
Incorrectly Classified Instances    10           4.8544 %
Kappa statistic                    0.8836
Mean absolute error                 0.0425
Root mean squared error             0.1265
Relative absolute error             19.4132 %
Root relative squared error         39.2451 %
Total Number of Instances          206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.917   0.014   0.965    0.917   0.940     0.917   0.997    0.992    '(-inf-95125]'
      0.966   0.083   0.966    0.966   0.966     0.882   0.992    0.997    '(95125-145050]'
      ?       0.015   0.000    ?       ?         ?       ?       ?       '(145050-194975]'
      ?       0.000   ?       ?       ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.  0.951   0.063   0.966    0.951   0.958     0.893   0.993    0.995

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
55  5  0  0 |  a = '(-inf-95125]'
 2 141  3  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

Custom with OneR:

```

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      182           88.3495 %
Incorrectly Classified Instances    24           11.6505 %
Kappa statistic                    0.7161
Mean absolute error                 0.0583
Root mean squared error            0.2414
Relative absolute error            26.6041 %
Root relative squared error        74.8939 %
Total Number of Instances         206

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.767   0.041   0.885    0.767   0.821     0.759   0.863    0.746    '(-inf-95125]'
      0.932   0.233   0.907    0.932   0.919     0.713   0.849    0.893    '(95125-145050]'
      ?       0.019   0.000    ?       ?         ?       ?       ?       '(145050-194975]'
      ?       0.000   ?       ?       ?         ?       ?       ?       '(194975-inf)'
Weighted Avg.  0.883   0.177   0.900    0.883   0.891     0.726   0.853    0.850

=== Confusion Matrix ===

  a   b   c   d  <-- classified as
46  14   0   0 |  a = '(-inf-95125]'
 6 136   4   0 |  b = '(95125-145050]'
 0   0   0   0 |  c = '(145050-194975]'
 0   0   0   0 |  d = '(194975-inf)'

```

Custom with MLP:

```

Time taken to build model: 2.47 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      199           96.6019 %
Incorrectly Classified Instances     7            3.3981 %
Kappa statistic                     0.9181
Mean absolute error                  0.0257
Root mean squared error              0.121
Relative absolute error              11.7332 %
Root relative squared error          37.5597 %
Total Number of Instances           206

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC   ROC Area  PRC Area  Class
          0.950    0.027    0.934    0.950    0.942    0.918  0.986    0.981    '(-inf-95125]'
          0.973    0.050    0.979    0.973    0.976    0.918  0.978    0.981    '(95125-145050]'
          ?        0.000    ?        ?        ?        ?      ?      ?      '(145050-194975]'
          ?        0.000    ?        ?        ?        ?      ?      ?      '(194975-inf)'
Weighted Avg.  0.966    0.043    0.966    0.966    0.966    0.918  0.980    0.981

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
57  3  0  0 |  a = '(-inf-95125]'
 4 142  0  0 |  b = '(95125-145050]'
 0  0  0  0 |  c = '(145050-194975]'
 0  0  0  0 |  d = '(194975-inf)'

```

5.2 – Analysis

With our five different attribute selection algorithms and four different model types, we trained and tested a total of 20 models. The tables below show model accuracy from the training and test sets respectively, based on both the attribute selection algorithm and the model used, as well as their true positive rates, false positive rates, and ROC area. The best-performing scores are highlighted in green, and the lowest-performing in red. We trained our models using 10-fold cross-validation on our training set, and tested our models with the 20% test set using WEKA's supplied test set feature.

Training - Using Cross-validation

	Naive Bayes	Random Forest	OneR	MLP
Correlation	72.5467	95.0935	86.0981	92.5234
OneR	70.9112	93.5748	86.0981	89.6028
Info Gain	70.4439	95.0935	86.0981	94.2757
CfsSubset	87.9673	88.7850	86.4486	88.4346
Custom	70.2103	94.5093	86.0981	90.0701

Testing - Test Set

	Naive Bayes	Random Forest	OneR	MLP
Correlation	70.8738	97.5728	88.3495	95.1456
OneR	66.9903	97.0874	88.3495	97.0874
Info Gain	67.4757	96.6019	88.3495	96.6019
CfsSubset	91.7476	94.1748	88.3495	92.2330
Custom	68.4466	95.1456	88.3495	96.6019

True Positive Rate

	Naive Bayes	Random Forest	OneR	MLP
Correlation	0.725	0.951	0.861	0.925
OneR	0.709	0.936	0.861	0.896
Info Gain	0.704	0.951	0.861	0.943
CfsSubset	0.880	0.888	0.864	0.884
Custom	0.702	0.945	0.861	0.901

False Positive Rate

	Naive Bayes	Random Forest	OneR	MLP
Correlation	0.146	0.077	0.232	0.115
OneR	0.139	0.101	0.232	0.175
Info Gain	0.146	0.077	0.232	0.089
CfsSubset	0.211	0.118	0.225	0.214
Custom	0.151	0.085	0.232	0.162

ROC Area

	Naive Bayes	Random Forest	OneR	MLP
Correlation	0.882	0.987	0.814	0.972
OneR	0.895	0.986	0.814	0.952
Info Gain	0.891	0.984	0.814	0.965
CfsSubset	0.915	0.939	0.820	0.920
Custom	0.890	0.987	0.814	0.947

Based on the results of training all 20 models, the Pearson Correlation approach appears to be the most effective method of predicting the class. Additionally, the best-performing model across all four attribute selection types is Random Forest, being the only type of model to score over 94% testing accuracy across all five attribute selection types, and has the best ROC area and lowest FP. Overall, the best accuracy achieved by a model was 97.5728% accuracy from the Random Forest model with the Correlation attribute selection. The model best suited for deployment overall is also the **Random Forest model with Correlation attribute selection**, having a TPR of 0.951, an FPR of 0.077, and an ROC area of 0.987. All of these metrics suggest that this model performs best in every way. As such, it would be best to choose this model as our model in the event of deployment.

One recurring theme with our model results is the fact that our testing accuracies were generally better, not worse, than the training accuracies. This can be attributed to one simple reason: our fourth bin of median income is too small. The entire dataset has a total of 1070 instances, with just three of them belonging to the highest income bracket. When it was time for the train-test

split, only one of them managed to go into the test set. As such, there may have been bias that may have inflated our results, because it is so poorly represented.

6. Conclusion/How to Reproduce our Model

The purpose of this project was to gain an understanding of how useful machine learning can be in predicting general practitioners' salaries, as well as the wider applications of predicting these salaries based on demographics. We were able to train and test twenty different machine learning models, all of which are capable of predicting GP salaries using demographic data with at least 66% accuracy. Our best-performing model was our Random Forest model with Correlation attribute selection, achieving an accuracy of 97.5728% on testing, as well as TPR, FPR, and ROC values of 0.951, 0.077, and 0.987 respectively. A major limitation of our project was the lack of representation with our labels, with just three out of 1070 instances belonging to the highest income tax bracket, negatively affecting predictability and artificially inflating accuracies without actually learning from it. Future directions for this project include gathering more data from high-income general practitioners, or even oversampling to make the bracket represented. Also, future non-machine-learning studies can make use of this demographic data to further investigate the root causes of discrepancies in GP salaries, rather than just correlation as our models suggest. As such, investigating the root causes is what will enable policy-makers to further develop and improve quality of life.

Steps to reproduce our model: Random Forest with Info Gain attribute selection:

1. Open Weka and load training.arff in the Train-Test-Datasets/Original-Dataset.
2. Go to the Select Attributes tab and choose the class - Median Income Before Tax.
3. Select InfoGainAttributeEval as the attribute evaluator and Ranker as the search method and hit Start.
4. Remove all features with a ratio of less than 0.05 and keep the remaining features and the class in the dataset.
5. Save this dataset as an arff.
6. Repeat steps 1-5 for testing.arff.
7. **The above files can be found under Train-Test-Datasets/Info-Gain**
8. Open Weka and load the training set.
9. Click on the classify tab and click "Supplied test set" under Test Options.
10. Load the testing dataset and select the correct class - Median Income Before Tax.
11. Select Random Forest as the classifier algorithm. This is under the trees folder.
12. Click Start.

13. The model can be found here:

Model_Performance_Data/Best_model_RandomForest.model

7. Team Members and Tasks Performed

7.1 Dataset Selection - Dev

7.2 Proposal - Dev, Anirudh

7.3 Preprocessing: Handling Missing Data - Dev

7.4 Preprocessing: Normalization - Anirudh

7.5 Attribute Selection Algorithms - Anirudh, Dev

7.6 Train-Test-Splits - Anirudh, Dev

7.7 Model Training, Cross Validation - Anirudh, Dev

7.8 Results - Dev

7.9 Analysis - Anirudh

7.10 Conclusion - Anirudh

7.11 Final Report - Dev, Anirudh