GP Earnings and Expenses Report: Median GP Earnings Model

Team Members:

Dev Makwana, Anirudh Chintaluri

10/20/2024Table of Contents

[**Part 1 – Statement/Project Goal 3**](#_5oqeonqo4dq0)

[**Part 2 – Description of Dataset 3**](#_loyc3321rkmp)

[**Part 3 – Pre-Processing 7**](#_mtn1d95j0zj4)

[**Part 4 – Attribute Selection Algorithms & Model Classifiers Used 8**](#_u3i7dwu9kr6e)

[**Part 5 – Results and Analysis 13**](#_vx3rjxz9vr8o)

[**Part 6 – Conclusion/How to Reproduce Our Model 32**](#_flxcly9bx7op)

[**Part 7 - Team Members and Tasks Performed 33**](#_r3hfg4gthv3s)

## 

## 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:

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 vaue of 0.05 to remove all attributes with correlation coefficients below the cutoff value. The attributes retained with this algorithm are below:



**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*

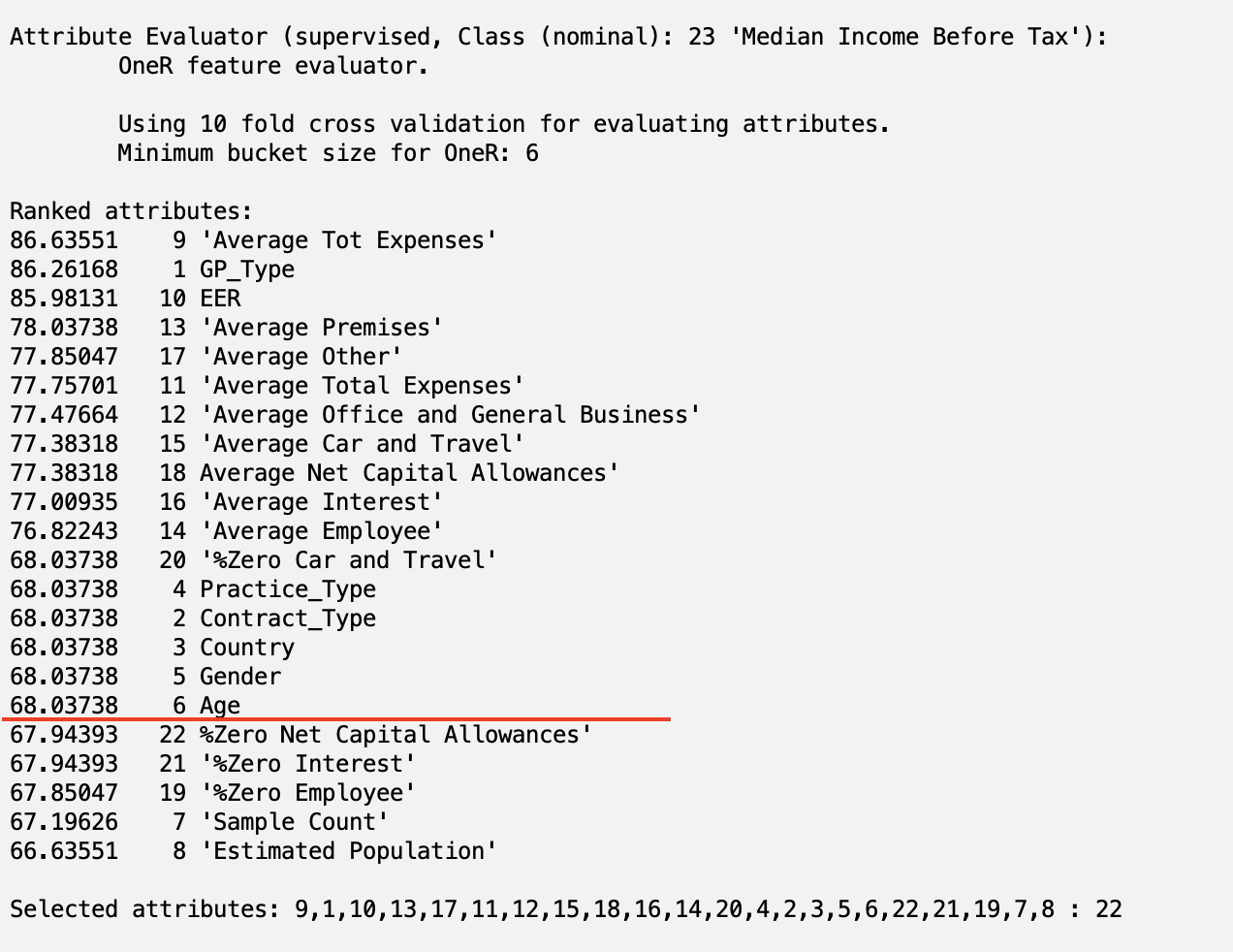
*find the most frequent class value*

*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.



**4.1.3 – InfoGainAttributeEval**

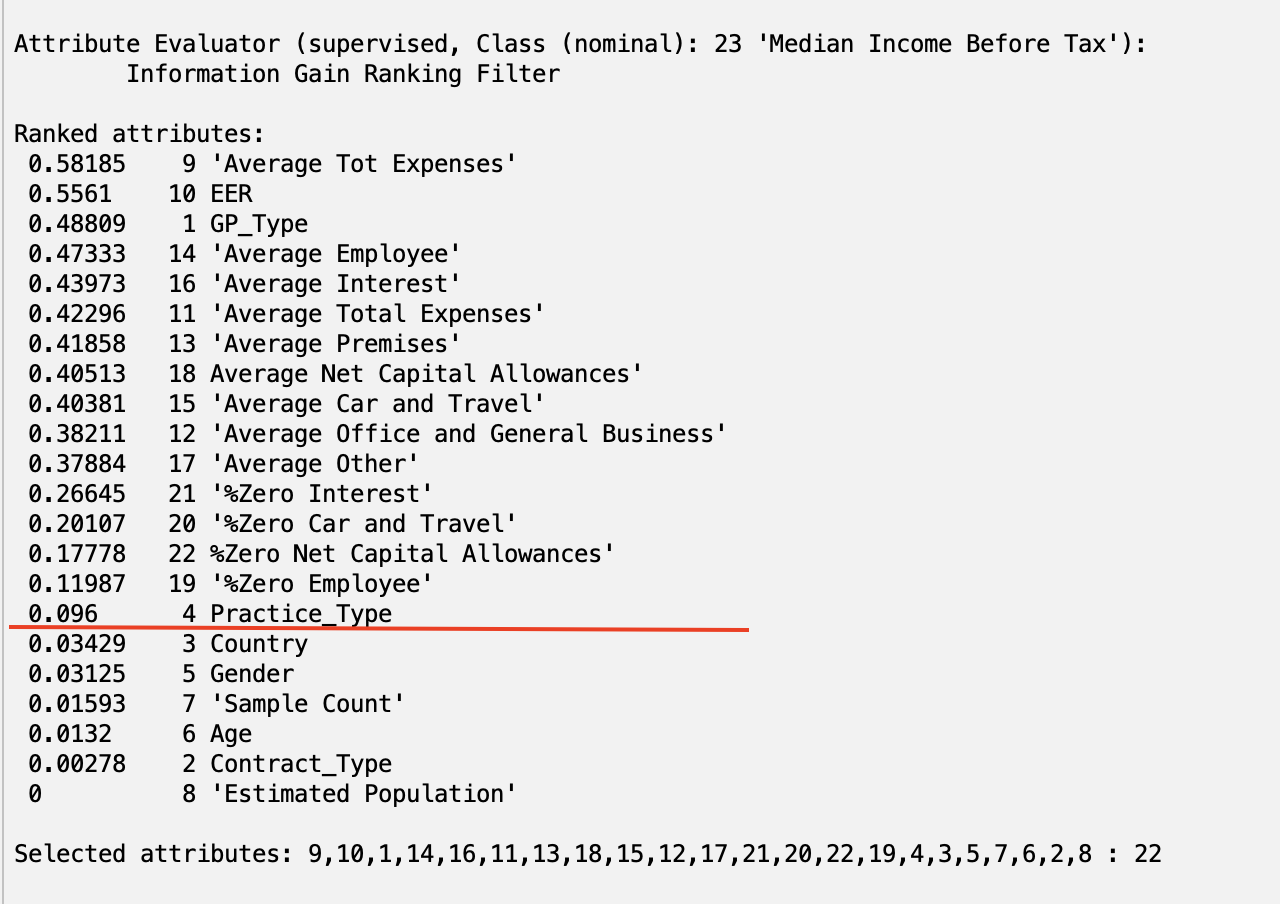
This algorithm uses the following formulas to calculate Gain(A) for each attribute A:

Gain(A) = Info(D) - InfoA(D)

Info(D) = - log2()

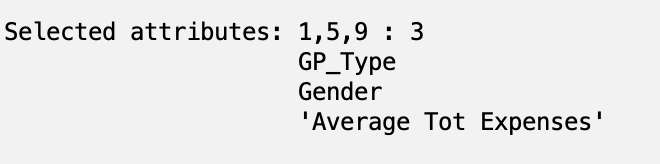
where is the probability that a tuple in D belongs to class Ci, estimated by and *m* is the number of classes. Info is the expected information needed to classify a tuple in D. After calculating the gain for each attribute, the attribute with the highest information gain is selected as the best attribute.

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



**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:



**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 xi representing the value of the attribute Ai

m classes represented by C1, C2, .., Cm

P(Ci | X) =

where P(C1) = (# yes tuples) / (total # tuples) and P(C2) = (# no tuples) / (total # tuples). Assuming that no attributes depend on any other attributes:

**4.2.2 – trees.RandomForest**

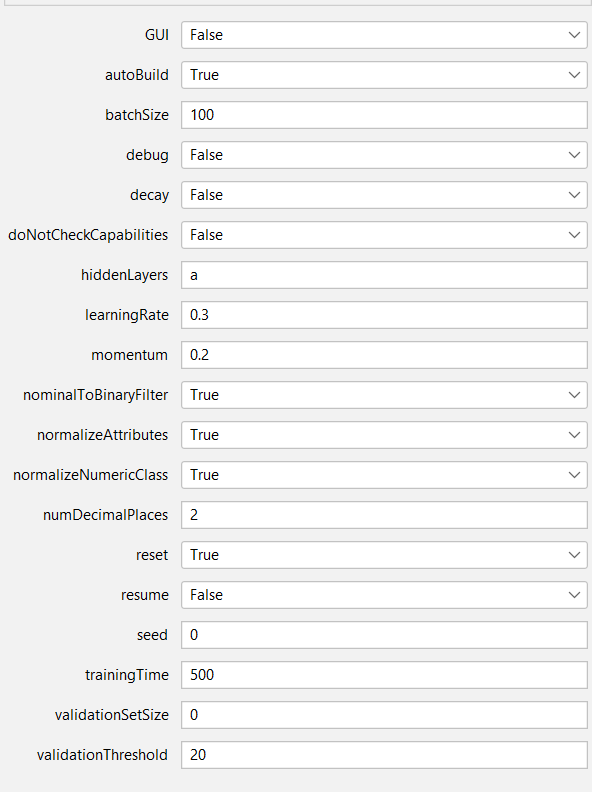
This voting-based classifier creates a forest of random trees. It consists of many separate decision trees that determine a class prediction, where the class with the highest number of trees for each tuple becomes the model’s prediction. All trees are equally weighted.

**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.



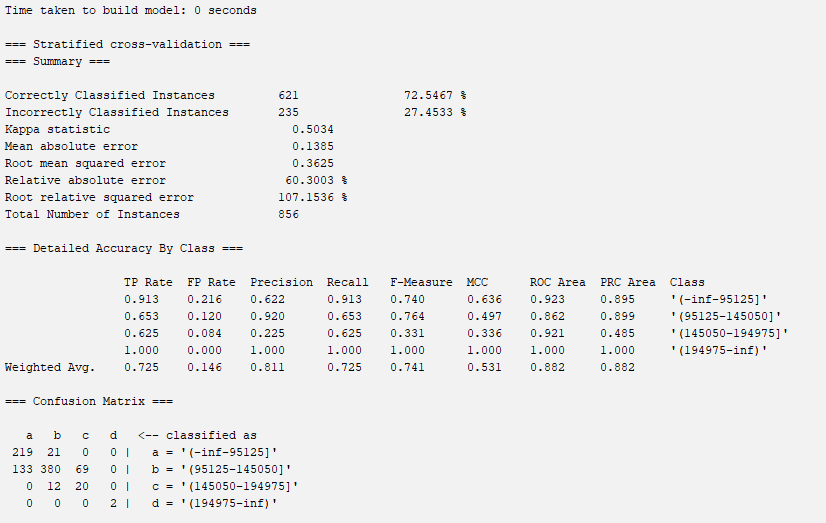
## 5. Results and Analysis

**5.1 – Results**

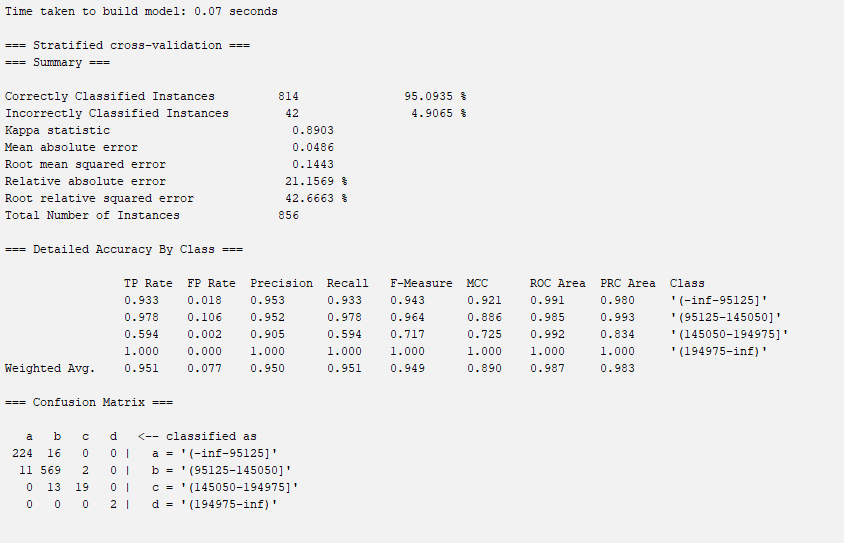
All classification algorithms were run using cross-validation with 10 folds and with the supplied testing set option.

**5.1.1 – Results using cross-validation**

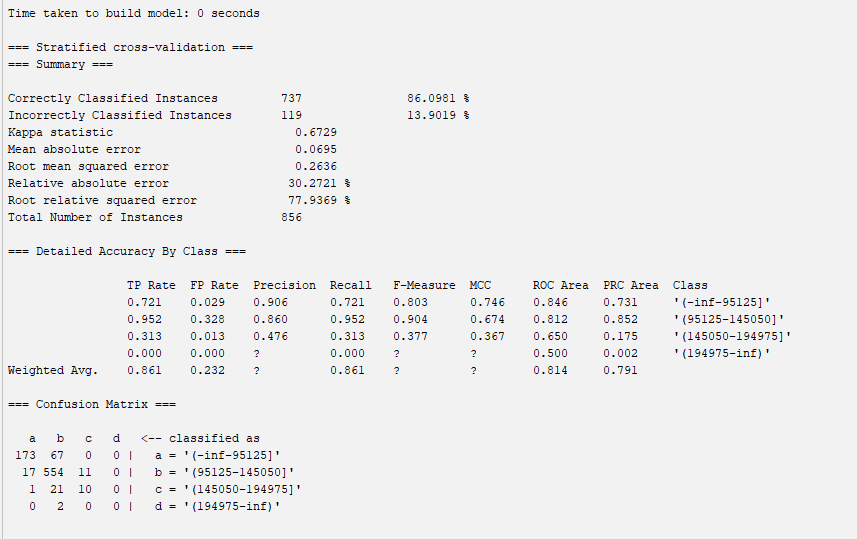
Correlation with Naive Bayes:



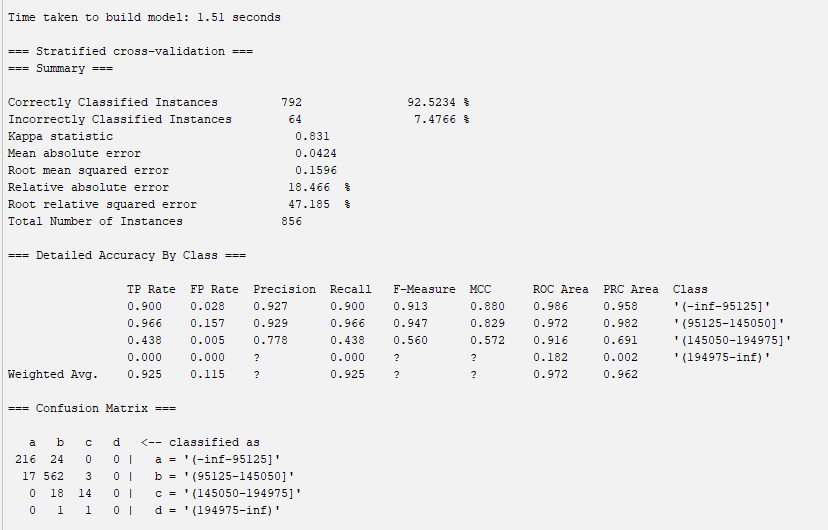
Correlation with Random Forest:



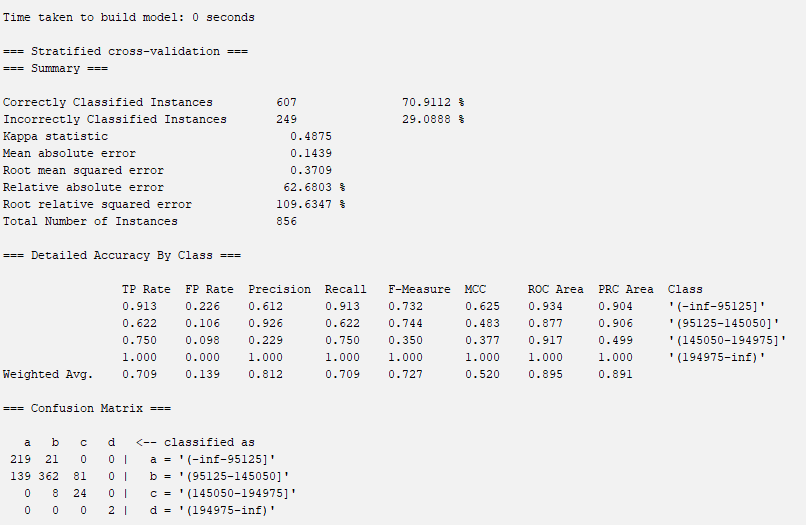
Correlation with OneR:



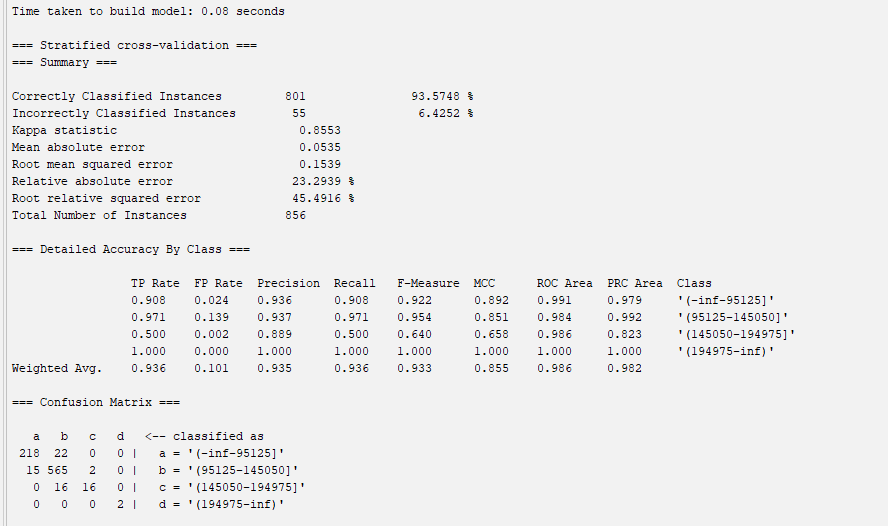
Correlation with MLP:



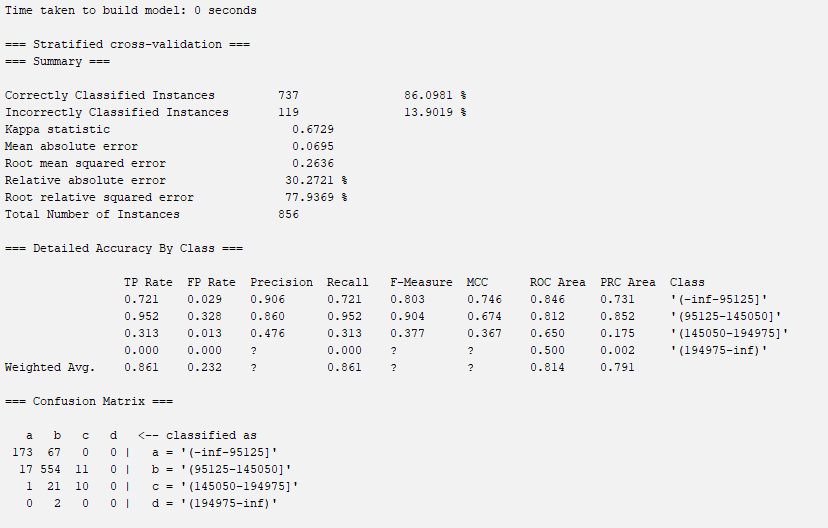
OneR with Naive Bayes:



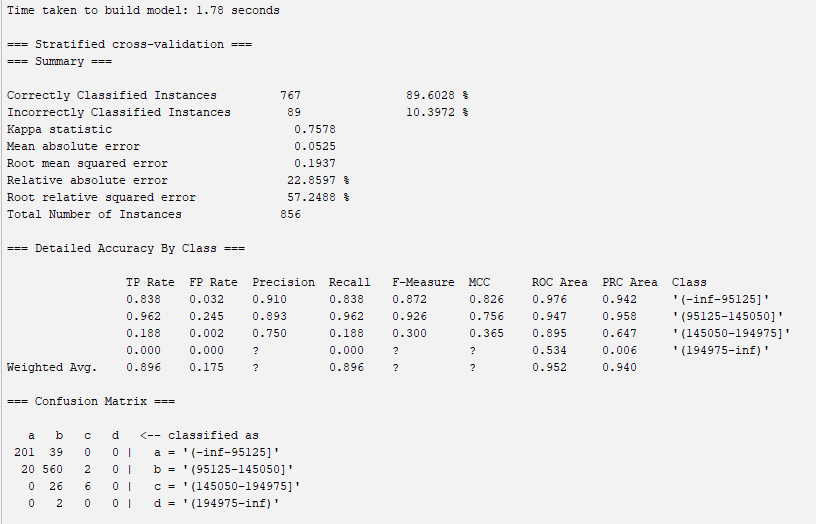
OneR with Random Forest:



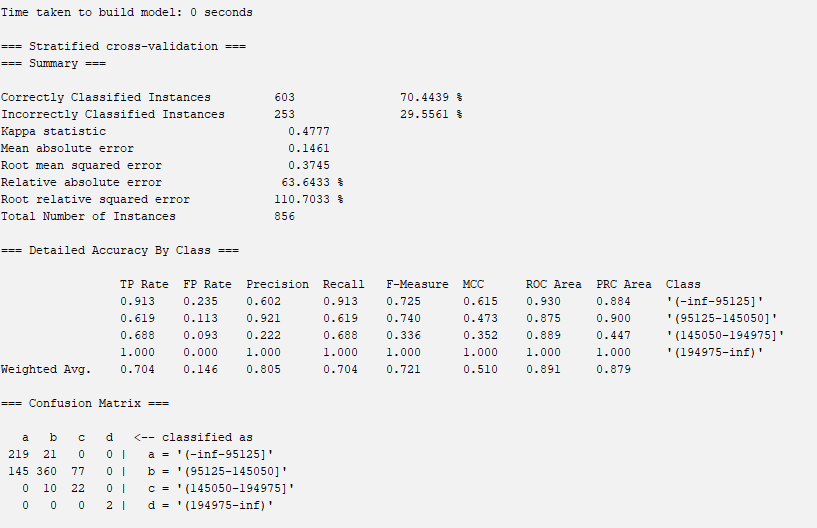
OneR with OneR:



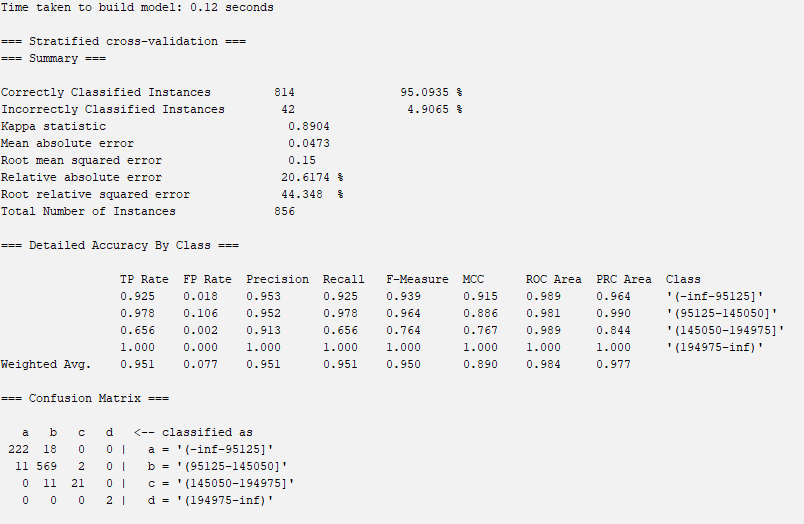
OneR with MLP:



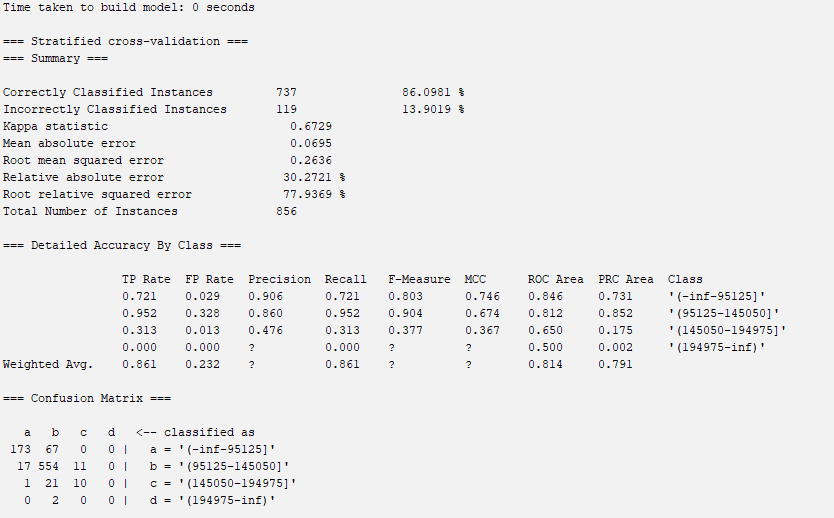
Info Gain with Naive Bayes:



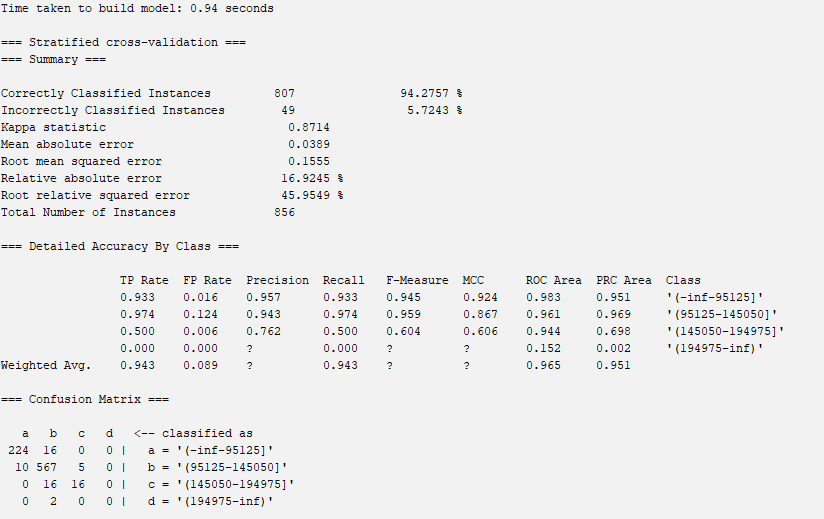
Info Gain with Random Forest:



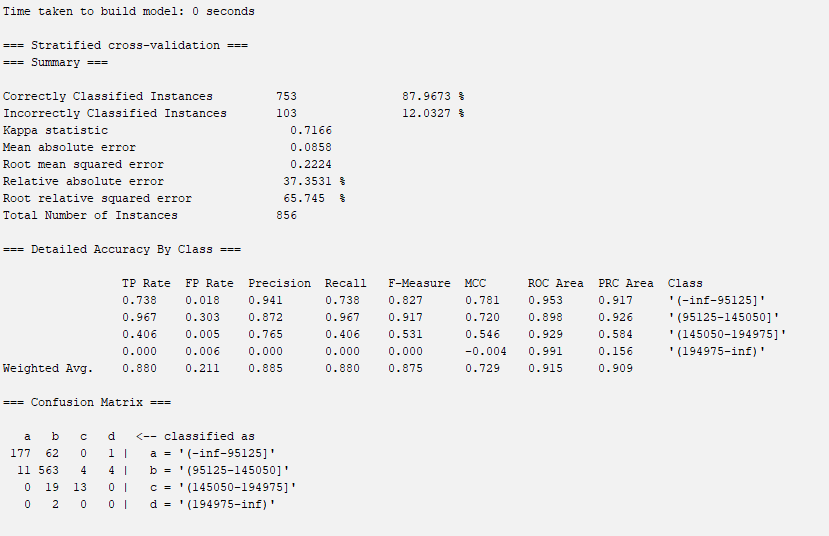
Info Gain with OneR:



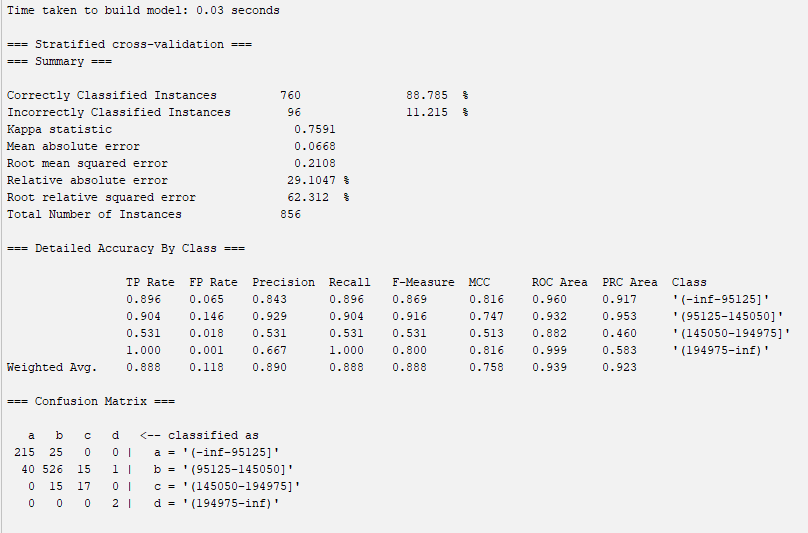
Info Gain with MLP:



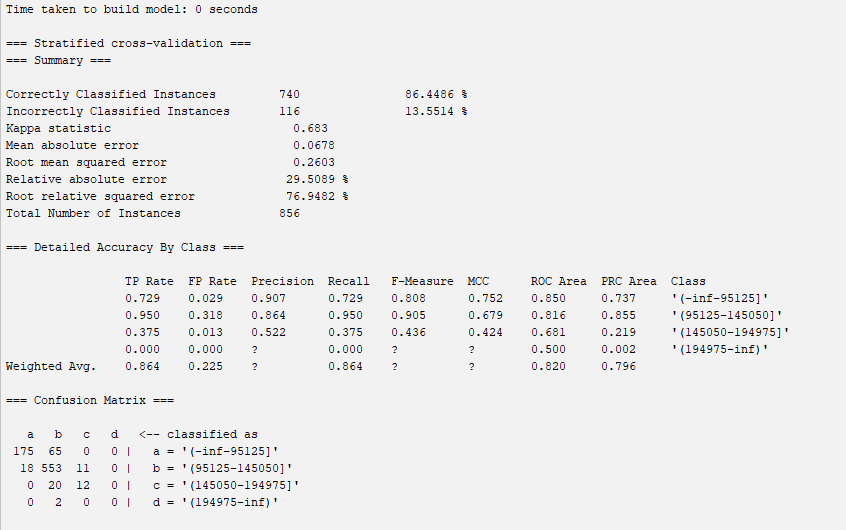
CfsSubset with Naive Bayes:



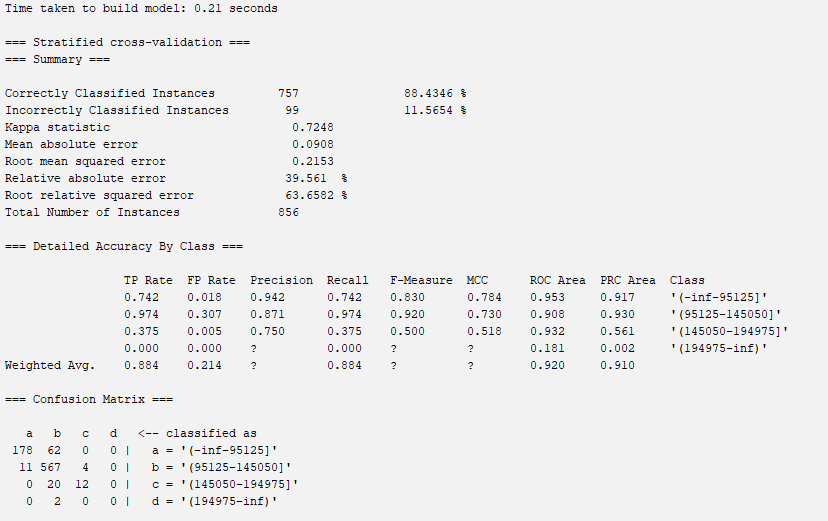
CfsSubset with Random Forest:



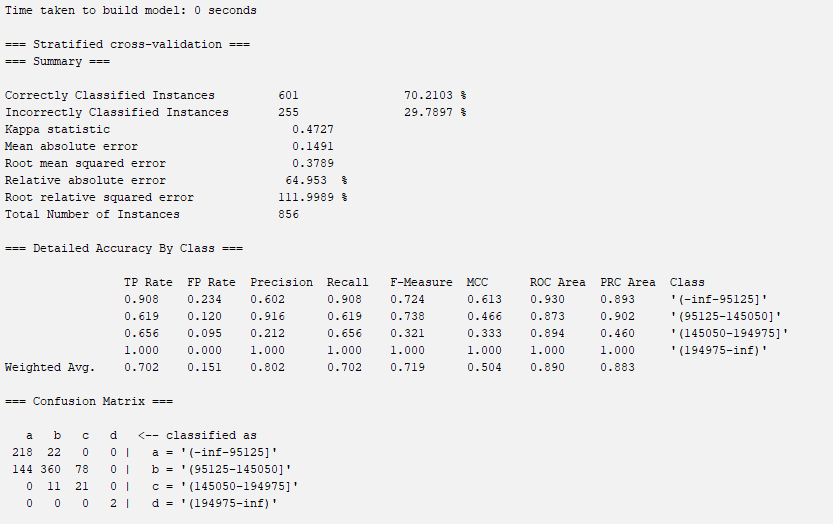
CfsSubset with OneR:



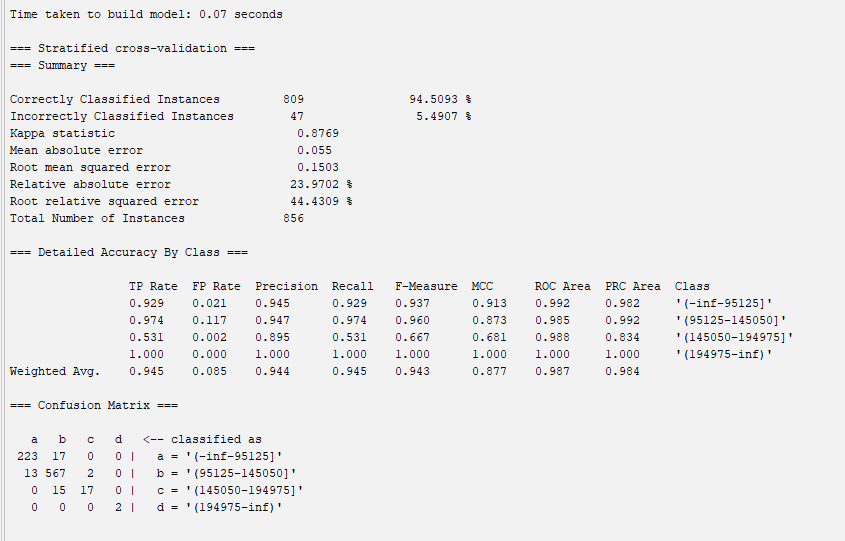
CfsSubset with MLP:



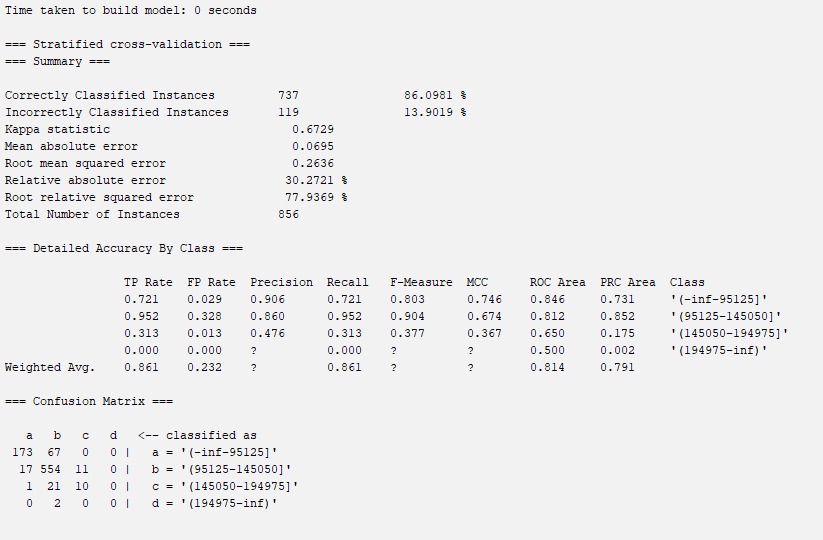
Custom with Naive Bayes:



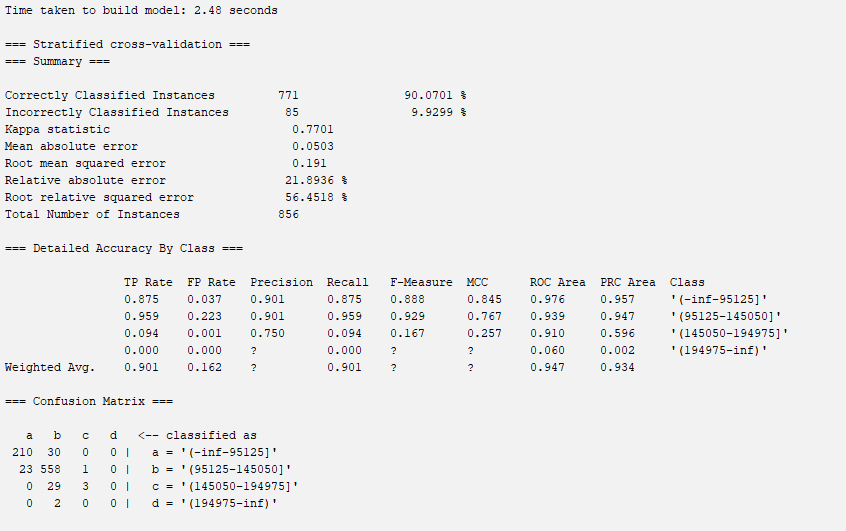
Custom with Random Forest:



Custom with OneR:

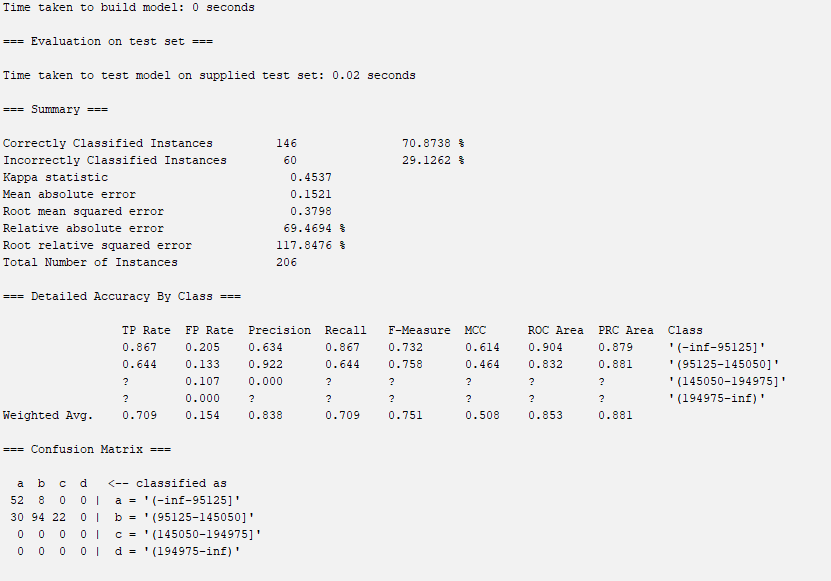


Custom with MLP:

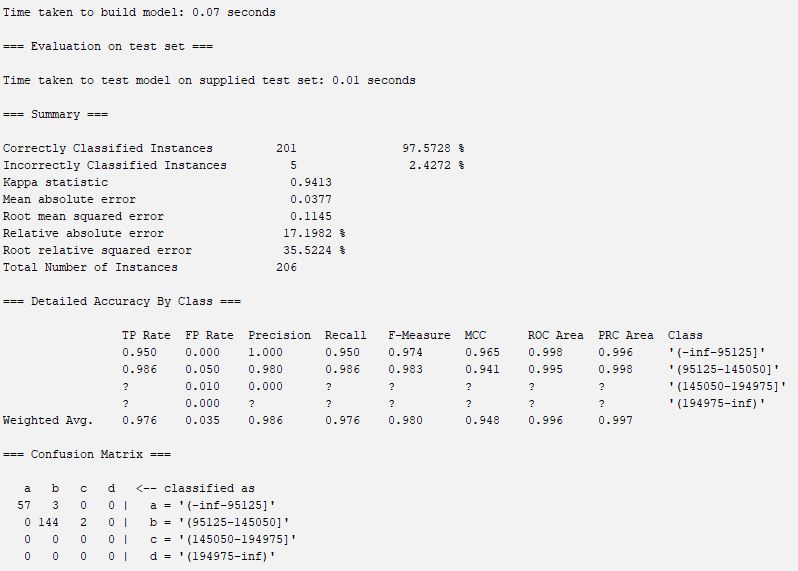


**5.1.2 – Results using test set**

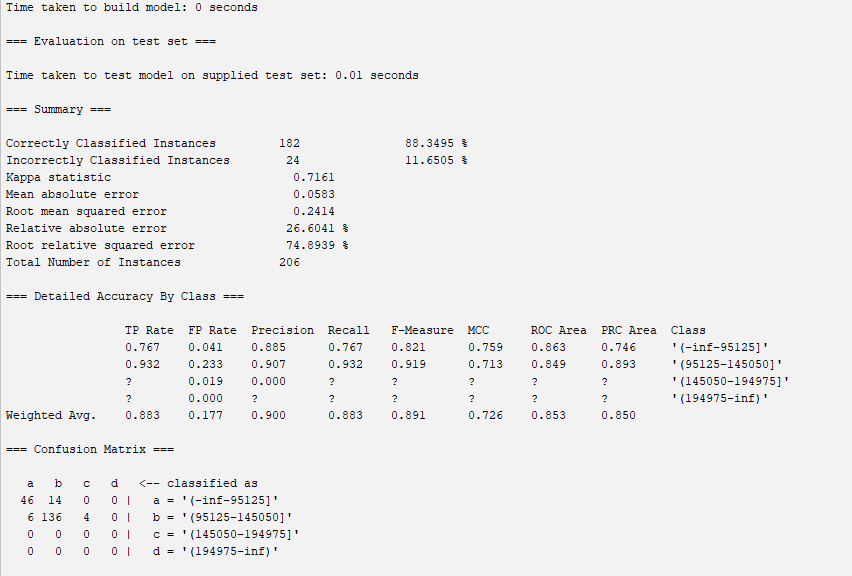
Correlation with Naive Bayes:



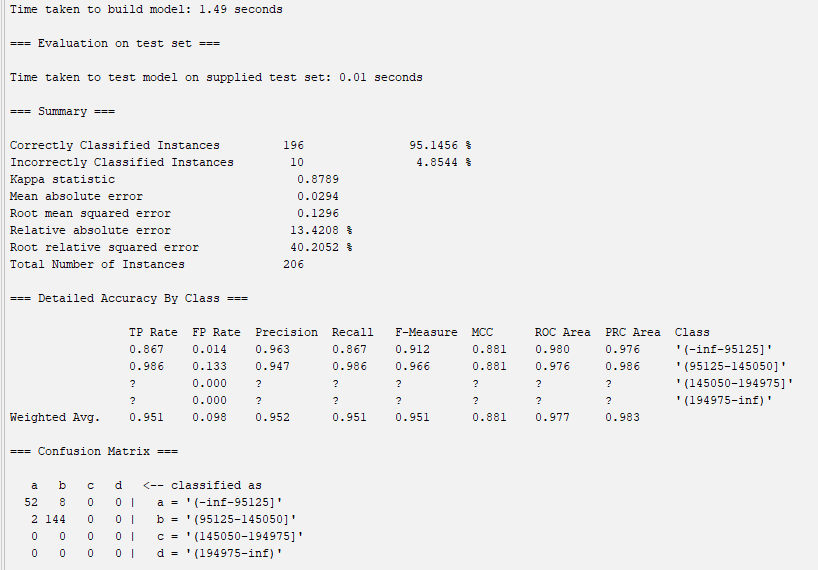
Correlation with Random Forest:



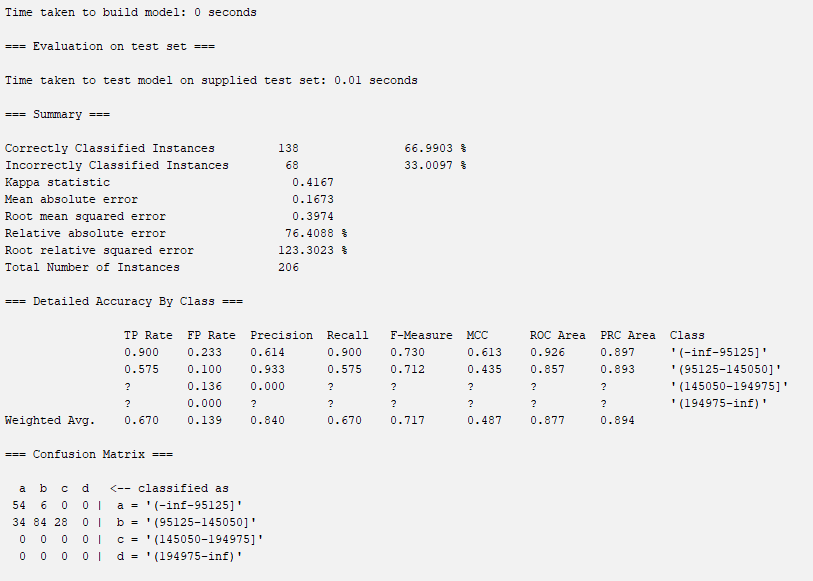
Correlation with OneR:



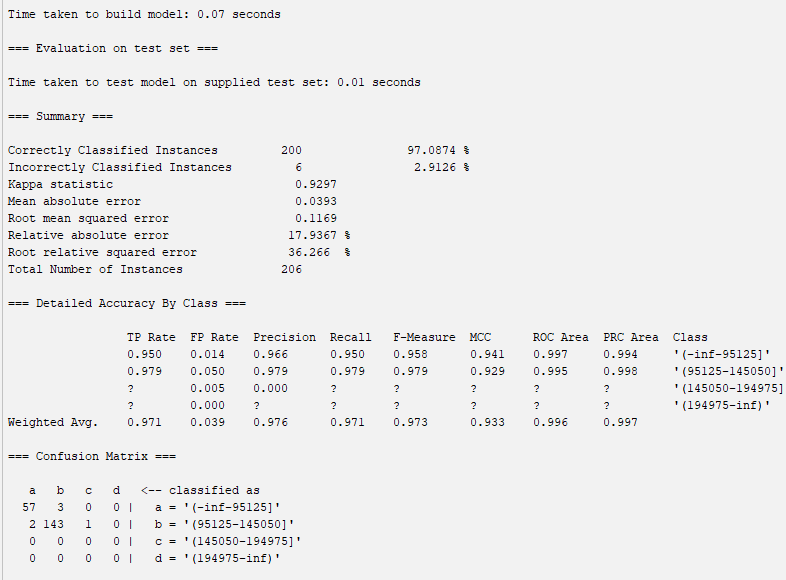
Correlation with MLP:



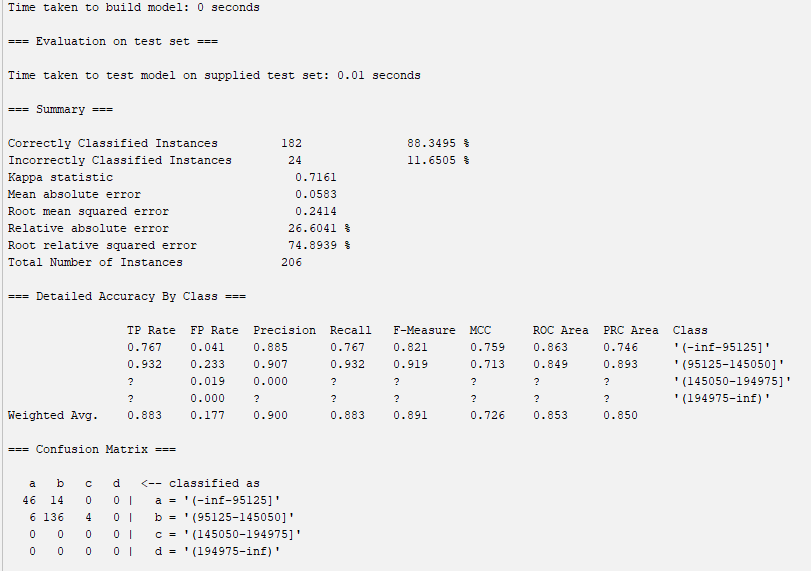
OneR with Naive Bayes:



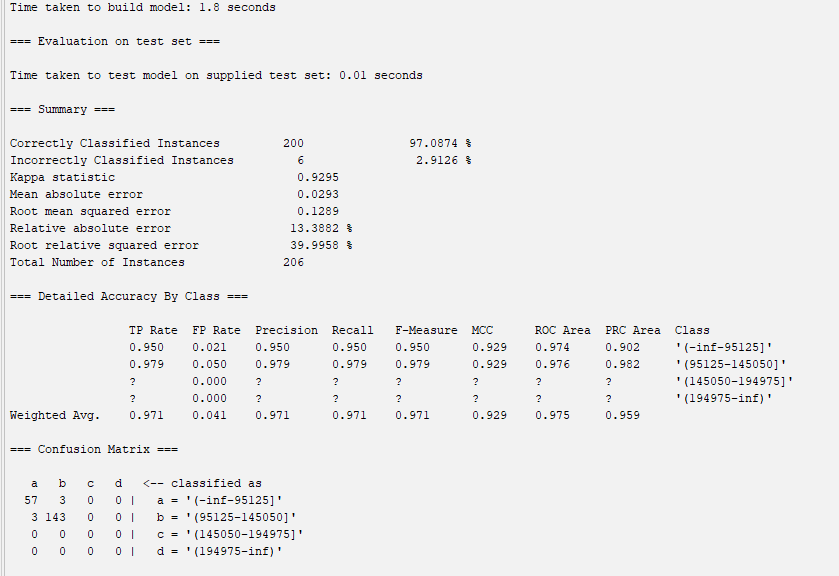
OneR with Random Forest:



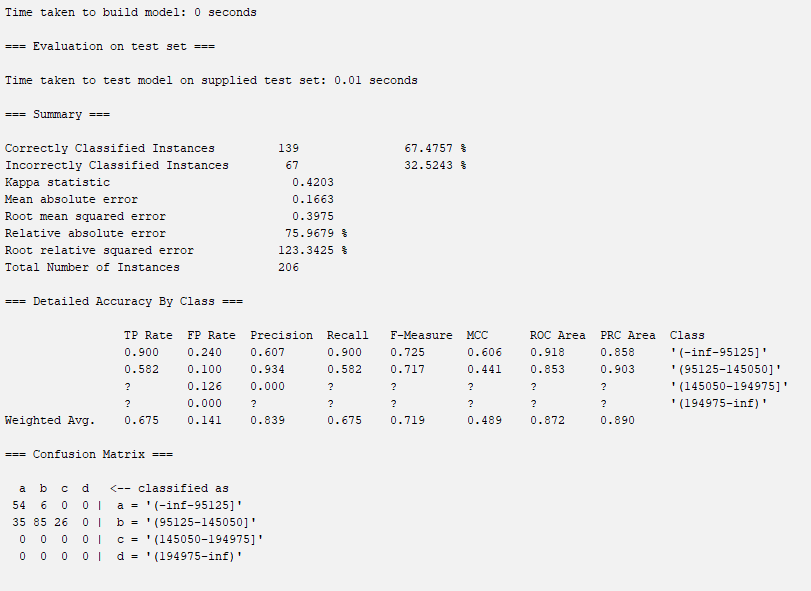
OneR with OneR:



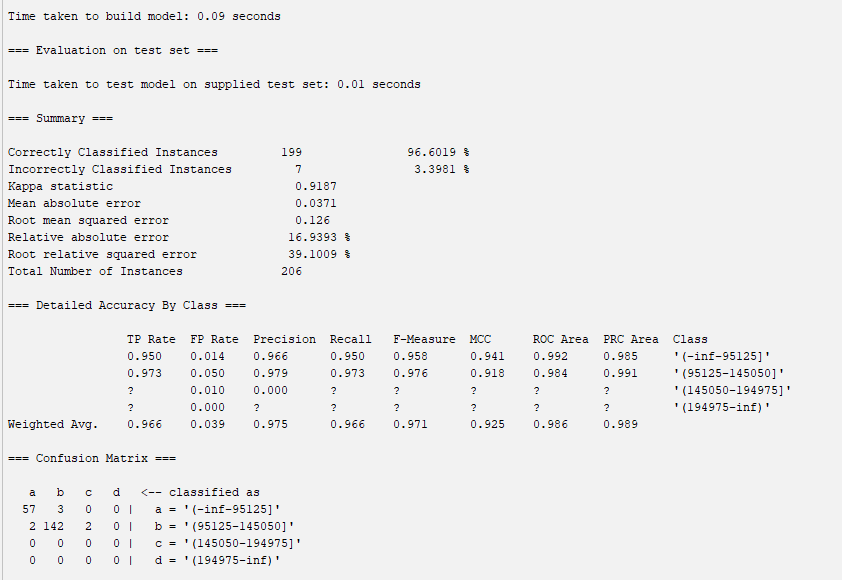
OneR with MLP:



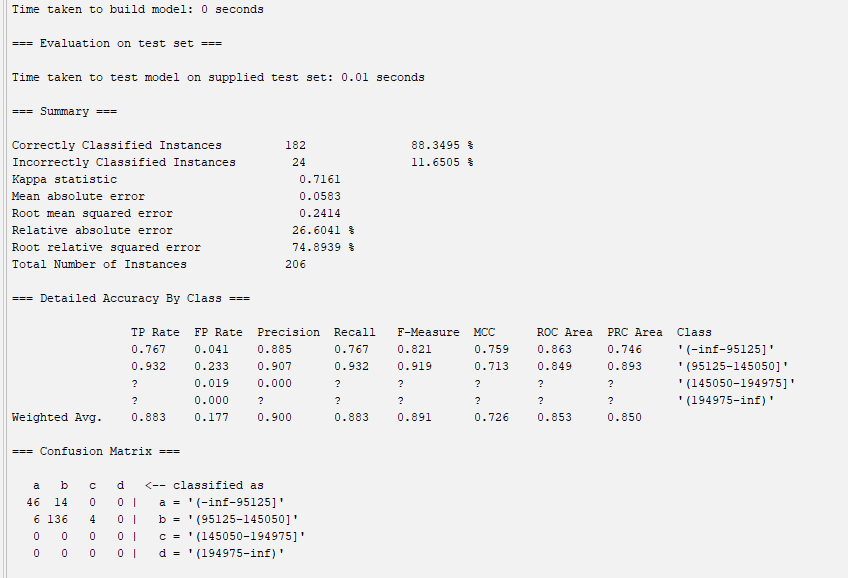
Info Gain with Naive Bayes:



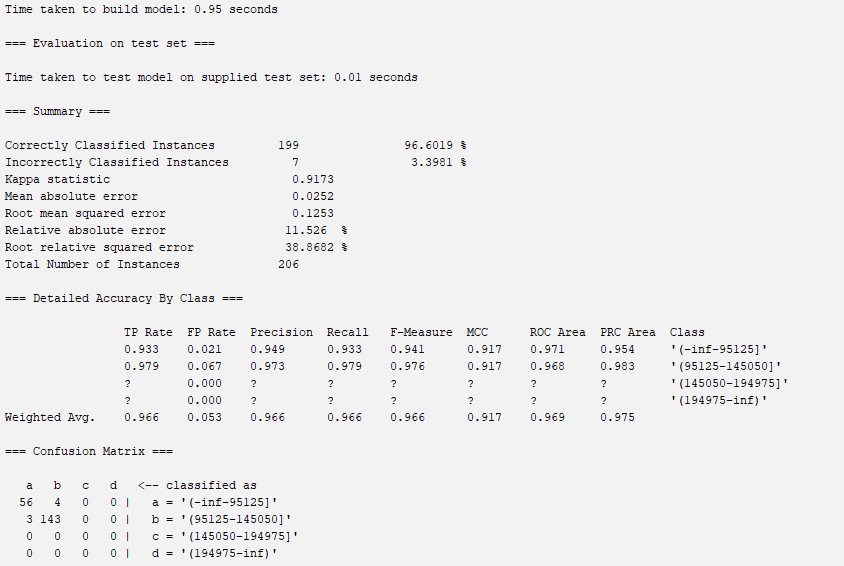
Info Gain with Random Forest:



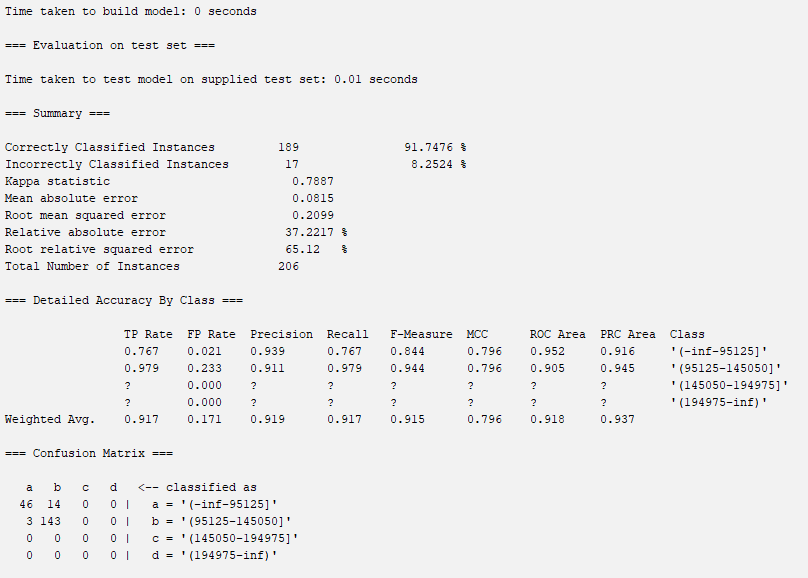
Info Gain with OneR:



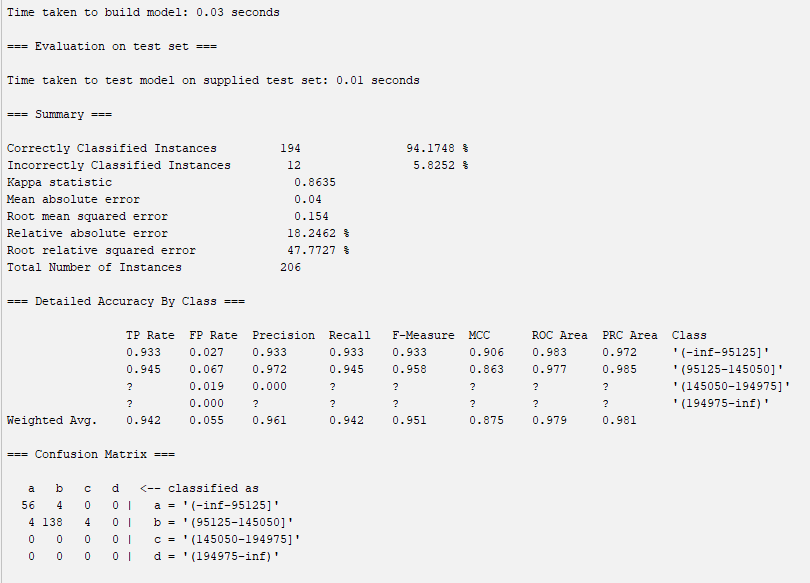
Info Gain with MLP:



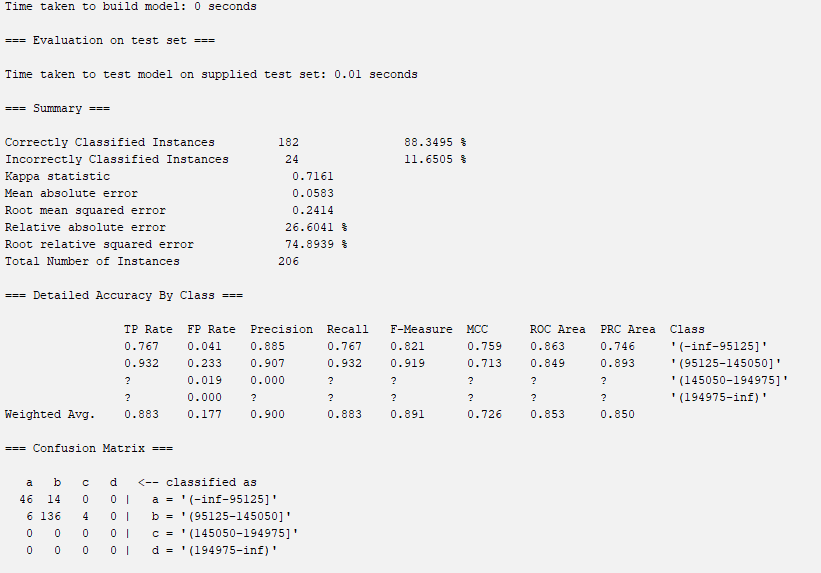
CfsSubset with Naive Bayes:



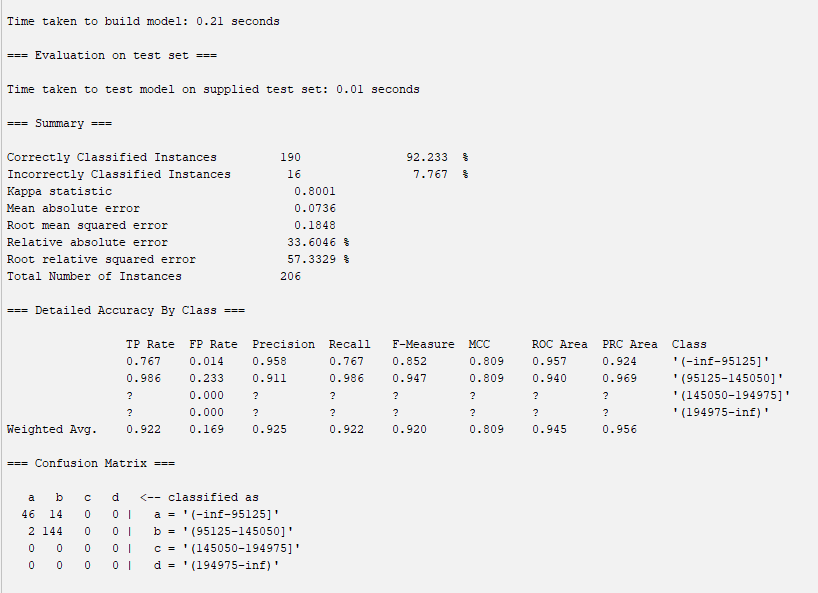
CfsSubset with Random Forest:



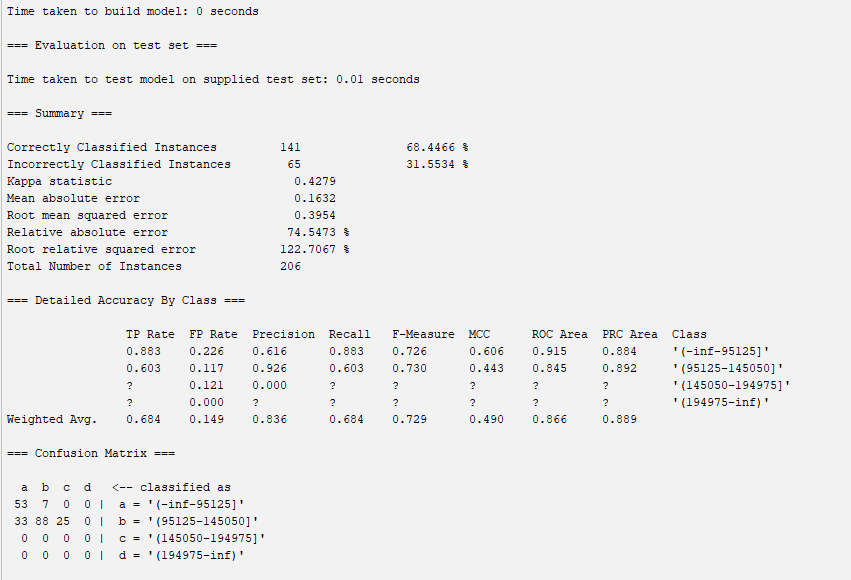
CfsSubset with OneR:



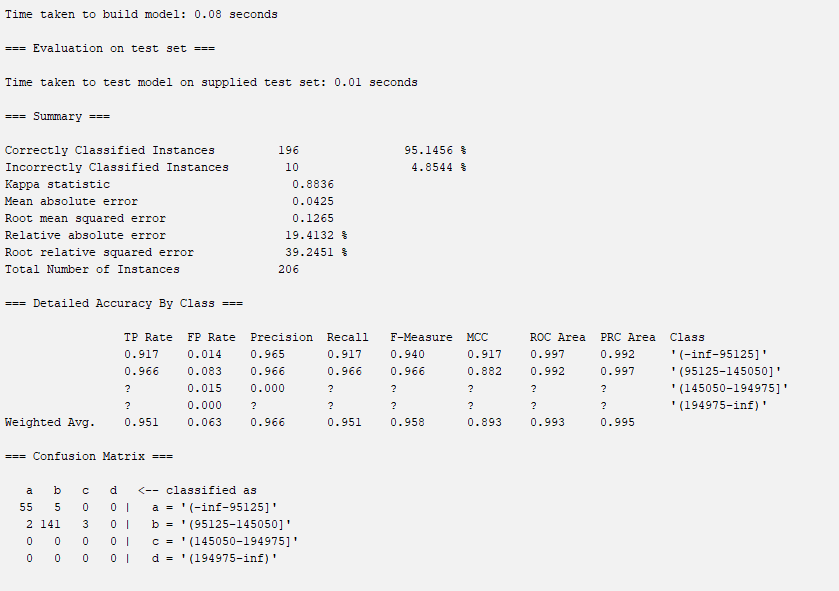
CfsSubset with MLP:



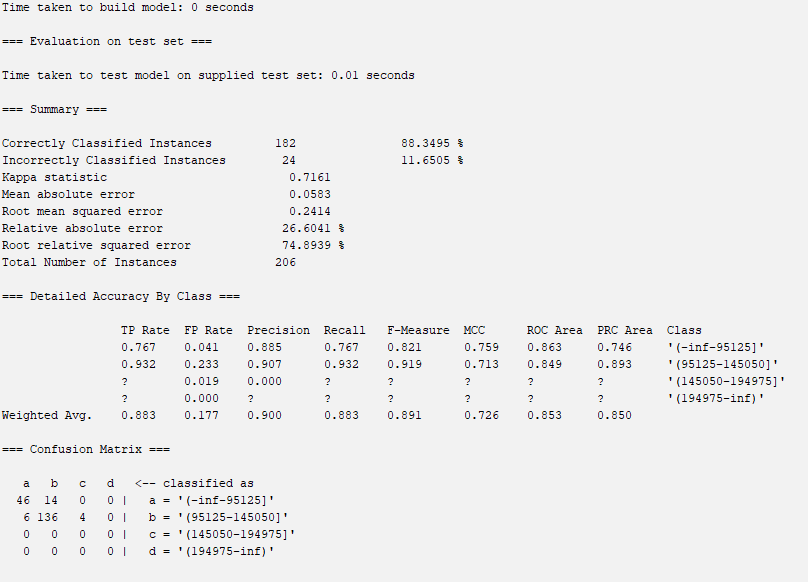
Custom with Naive Bayes:



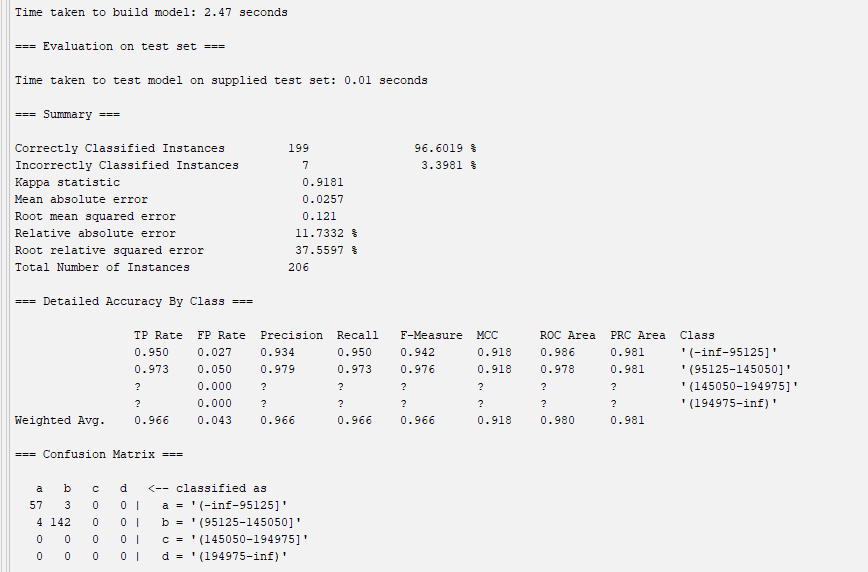
Custom with Random Forest:



Custom with OneR:



Custom with MLP:



**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 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**