### **MET CS699 SO1 DATA MINING**

### **FINAL PROJECT REPORT**

**Data Use**

project-2018-BRFSS-arthritis.arff

**Tool Use**

Weka: Using Weka to analynize the dataset

Excel: Using Excel to do the comparation to choose best model

**Preprocess**

Remove Missing Value: Before I proceeded with data processing, I checked the data and found many "?" in the dataset. These "?" had a significant impact on my output results. Therefore, I preprocessed the dataset by removing all data rows containing "?".

Data Reduction: For this dataset, there are about 108 attributes. But, not all attributes are useful. So I do the data reduction for this dataset, and make it remaining 22 attribute by different attribute selection methods.

**Attribute Selection Methods**

*CfsSubsetEval:*

CfsSubsetEval is a feature selection method used in machine learning. This method, standing for "Correlation-based Feature Selection", operates on the premise that a good feature subset contains features highly correlated with the classification, yet uncorrelated with each other. The CfsSubsetEval evaluation function measures the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy that exists among them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. This method has been implemented in popular machine learning tools like Weka, and it's often used in the preprocessing stage of a machine learning pipeline, helping to reduce dimensionality and improve model performance.

*CorrelationAttributeEval:*

CorrelationAttributeEval is a feature selection technique used in machine learning, specifically implemented in the Weka machine learning library. The method evaluates the worth of an attribute by measuring the correlation between it and the class. The correlation metric used can be either Pearson's correlation, which measures linear relationships, or Spearman's correlation, which measures monotonic relationships. The absolute value of the correlation signifies the strength of the relationship with the class, with 1 being a perfect correlation, -1 a perfect inverse correlation, and 0 no correlation. In the context of feature selection, this method can be used to rank attributes by their relevance to the class. It's a univariate filter method, meaning it considers each attribute independently of the others. Attributes with high absolute correlation values are considered more important for prediction tasks. This method is often used in the preprocessing stage of a machine learning pipeline to select features that have the most predictive power, thereby reducing the dimensionality of the dataset and potentially improving the performance of the model.

*InfoGainAttributeEval*

InfoGainAttributeEval is a feature selection method used in machine learning and specifically implemented in the Weka machine learning library. This method evaluates the worth of an attribute by measuring the information gain in relation to the class.Information gain is a metric that measures how much information is gained about the class by knowing the value of an attribute. It's based on the concept of entropy from information theory, which quantifies the uncertainty or impurity in a set of instances. The information gain of an attribute is calculated as the difference in entropy before and after the attribute is given. In the context of feature selection, this method can be used to rank attributes by their information gain values. Attributes with high information gain are considered more important for prediction tasks because they provide more useful information about the class.This method is often used in the preprocessing stage of a machine learning pipeline to select features that have the most predictive power. By reducing the dimensionality of the dataset and focusing on the most informative attributes, InfoGainAttributeEval can help improve the performance of the model.

*OneRAttributeEval*

OneRAttributeEval is a feature selection method used in machine learning, specifically implemented in the Weka machine learning library. This method evaluates the worth of an attribute by using the OneR (One Rule) algorithm, which generates a single rule for each attribute in the data and then selects the rule with the smallest total error rate. The OneR algorithm, or "One Rule", is a simple, yet often surprisingly accurate, classification algorithm that generates one rule for each predictor in the data, then selects the one rule with the smallest total error.In the context of feature selection, this method can be used to rank attributes based on the accuracy of their corresponding OneR rules. Attributes whose OneR rules have low error rates are considered more important for prediction tasks.This method is often used in the preprocessing stage of a machine learning pipeline to select features that have the most predictive power. By focusing on the most informative attributes,OneRAttributeEval can help improve the performance of the model.

*SymmetricalUncertAttributeEval*

SymmetricalUncertAttributeEval is a feature selection method used in machine learning, specifically implemented in the Weka machine learning library. This method evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.Symmetrical Uncertainty is a measure derived from Information Theory. It is a symmetric measure of association between two random variables, providing a normalized value in the range of 0 (no association between the variables) to 1 (complete association). It is based on the concept of entropy and information gain.In the context of feature selection, SymmetricalUncertAttributeEval can be used to rank attributes based on their symmetrical uncertainty values in relation to the class. Attributes with high symmetrical uncertainty are considered more important for prediction tasks because they share a strong association with the class.

This method is often used in the preprocessing stage of a machine learning pipeline to select features that have the most predictive power. By reducing the dimensionality of the dataset and focusing on the most informative attributes, SymmetricalUncertAttributeEval can help improve the performance of the model.

**Attributes After Attribute Selections(Order by Relevant)**

*CfsSubsetEval:*

|  |  |  |  |
| --- | --- | --- | --- |
| *employ1* | *pneuvac4* | *diffwalk* | *diffdres* |
| *diabete3* | *physhlth* | *chckdny1* | *iday* |
| *persdoc2* | *checkup1* | *chcocncr* | *addepev2* |
| *chcscncr* | *cvdstrk3* | *qstver* | *x.age80* |
| *x.ageg5yr* | *x.chldcnt* | *x.rfbing5* | *x.exteth3* |
| *x.casthm1* | *havarth3* |  |  |

*CorrelationAttributeEval*

|  |  |  |  |
| --- | --- | --- | --- |
| x.ageg5yr | x.age80 | checkup1 | *diffwalk* |
| x.rfbing5 | *pneuvac4* | *physhlth* | *employ1* |
| *x.exteth3* | x.phys14d | x.rfhlth | *diabete3* |
| x.age65yr | x.hcvu651 | x.bmi5 | addepev2 |
| persdoc2 | chcscncr | flushot6 | x.age.g |
| x.totinda | exerany2 |  |  |

*InfoGainAttributeEval*

|  |  |  |  |
| --- | --- | --- | --- |
| *iday* | *diffwalk* | x.age80 | *x.age.g* |
| x.phys14d | *qstver* | x.ageg5yr | *genhlth* |
| *imonth* | *pneuvac4* | *physhlth* | checkup1 |
| x.rfbing5 | persdoc2 | *x.incomg* | *employ1* |
| *rmvteth4* | *x.chldcnt* | *diabete3* | *x.exteth3* |
| *chcscncr* | *x.rfhlth* |  |  |

*OneRAttributeEval*

|  |  |  |  |
| --- | --- | --- | --- |
| x.state | wtkg3 | *genhlth* | dispcode |
| iyear | hlthpln1 | menthlth | *physhlth* |
| rmvteth4 | lastden4 | x.psu | fmonth |
| imonth | *iday* | qstlang | x.metstat |
| cvdinfr4 | x.casthm1 | sleptim1 | persdoc2 |
| exerany2 | chcocncr |  |  |

*SymmetricalUncertAttributeEval*

|  |  |  |  |
| --- | --- | --- | --- |
| x.ageg5yr | x.age80 | *diffwalk* | x.rfbing5 |
| checkup1 | persdoc2 | *chcscncr* | *physhlth* |
| *x.age.g* | x.phys14d | *x.chldcnt* | *pneuvac4* |
| *employ1* | diffdres | *diabete3* | chckdny1 |
| cvdstrk3 | *x.exteth3* | x.rfhlth | addepev2 |
| genhlth | qstver |  |  |

**Classification Algorithm**

*NaiveBayes:* This is a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. It's particularly suited for high-dimensional datasets and is commonly used in natural language processing and spam filtering.

*Bagging :*Short for Bootstrap Aggregating, Bagging is a general-purpose procedure for reducing the variance of a machine learning method. It works by creating multiple subsets of the original dataset (with replacement), training a separate model on each subset, and then averaging the predictions (for regression) or voting for the most popular class (for classification).

*Logistic:*Logistic Regression is a statistical model that uses a logistic function to model a binary dependent variable. In machine learning, it's often used for binary classification problems - predicting one of two possible outcomes such as 'yes' or 'no'.

*Kstar :*K\* (K Star) is an instance-based classifier, which means it doesn't explicitly build a model. Instead, it memorizes the training instances which are subsequently used as knowledge for the prediction phase. The Kstar algorithm uses entropy-based distance measure.

*MultipleClassClassifier:*This refers to a classification problem with more than two classes. Each sample can only be labeled as one class. For instance, predicting the type of fruit (apple, orange, banana, etc.) based on certain features is a multi-class classification problem. Various algorithms can be used to solve this problem, including the ones mentioned above, as well as others like Decision Trees, Random Forests, and SVMs.

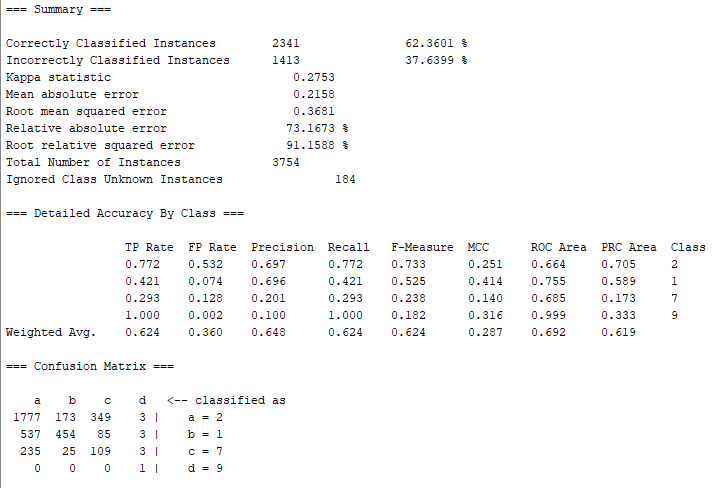
**What I learned**

Through the process of building these 25 models, I learned a great deal about the performance of different feature selection methods and machine learning models on my dataset. Here's what stood out to me: Effectiveness of Feature Selection Methods: I observed that the feature selection methods had a significant impact on the performance of the models. InfoGainAttributeEval seemed to consistently result in high accuracy across different models, indicating its effectiveness for this dataset. On the contrary, OneRAttributeEval consistently resulted in lower accuracy, suggesting that the one-rule heuristic might not be the best approach for feature selection with this particular data. Consistency of Certain Models: Certain models, notably Logistic and Bagging, performed well regardless of the feature selection method used. This consistency suggests that these models might be more robust or better suited to the characteristics of my dataset. Model-Specific Synergies: There were instances where specific combinations of models and feature selection methods stood out. For instance, when I combined Bagging with InfoGainAttributeEval, the model achieved 100% accuracy. This suggests a potential synergy between this specific model and feature selection method. Risk of Overfitting: I noticed that some models achieved perfect or near-perfect accuracy, such as the Bagging and Logistic models when paired with InfoGainAttributeEval. While at first glance this might seem like an ideal outcome, it also raised concerns about potential overfitting. These models might be overly tailored to the training data and could perform poorly when introduced to new, unseen data. Sensitivity of Models to Feature Selection: The variation in model performance across different feature selection methods provided valuable insights. For example, NaiveBayes showed a substantial range in performance, from just 38.192% accuracy with OneRAttributeEval to 99.5937% with InfoGainAttributeEval. This suggests that the choice of feature selection method can heavily influence the performance of certain models. As for the most interesting models, I was particularly intrigued by those that yielded the highest accuracy, such as Bagging and Logistic models with InfoGainAttributeEval. Additionally, models that showed significant differences in performance depending on the feature selection method, like NaiveBayes, offered fascinating insights into the sensitivity of models to feature selection methods. In the appendix of my report, I will provide detailed information on all 25 models, including the feature selection method used, the accuracy achieved, and other relevant metrics and observations, such as precision, recall, F1-score, training time, complexity, and so forth.

**Models Detail**

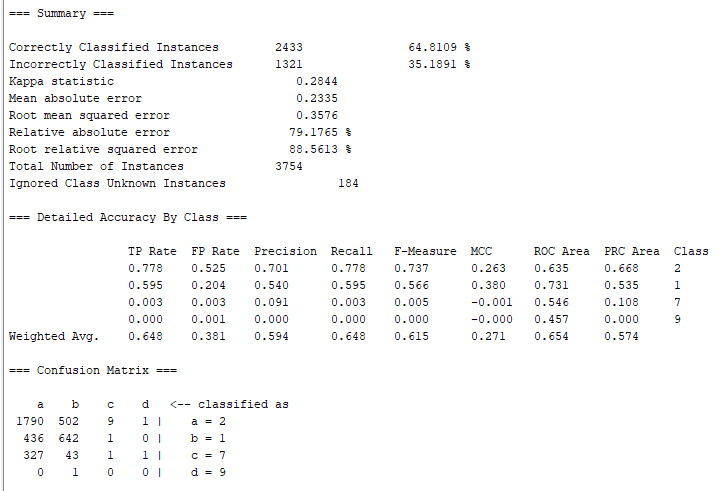
1. **CfsSubsetEval**
2. *NaiveBayes*

Accuracy: 62.3601%



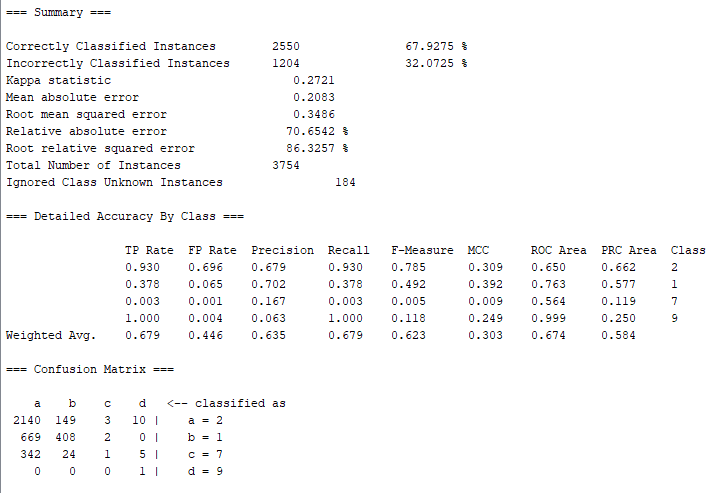
1. *Bagging*

Accuracy: 64.8109%

**

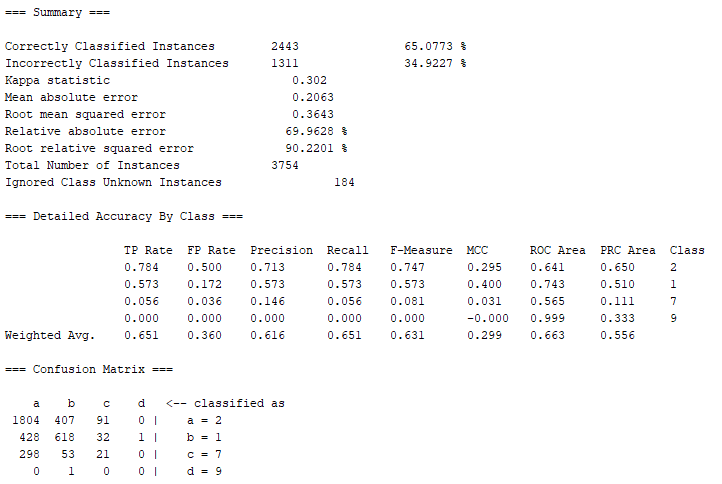
1. *Logistic*

Accuracy: 67.9275%

**

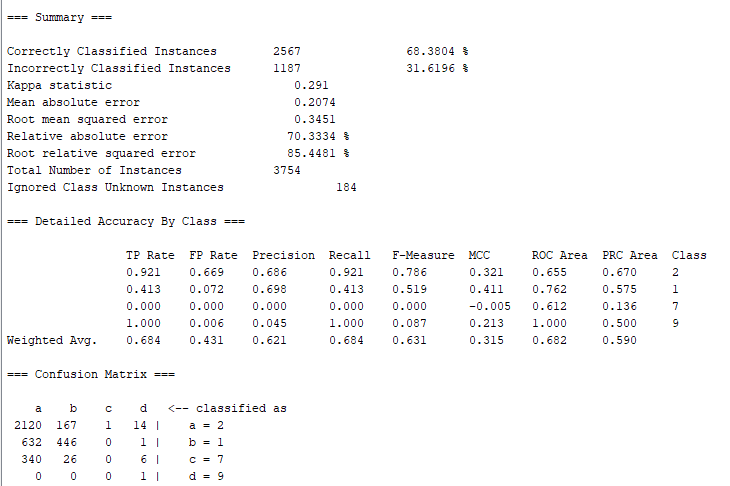
1. *Kstar*

Accuracy: 65.0773%

**

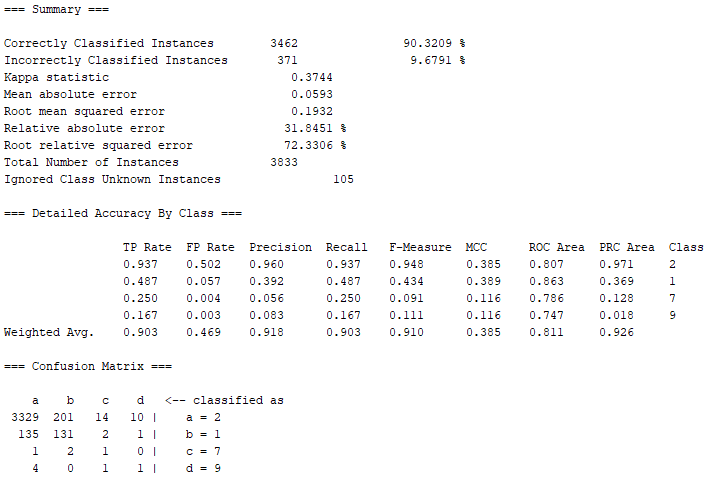
1. *MultipleClassClassifier*

Accuracy: 68.3804%

**

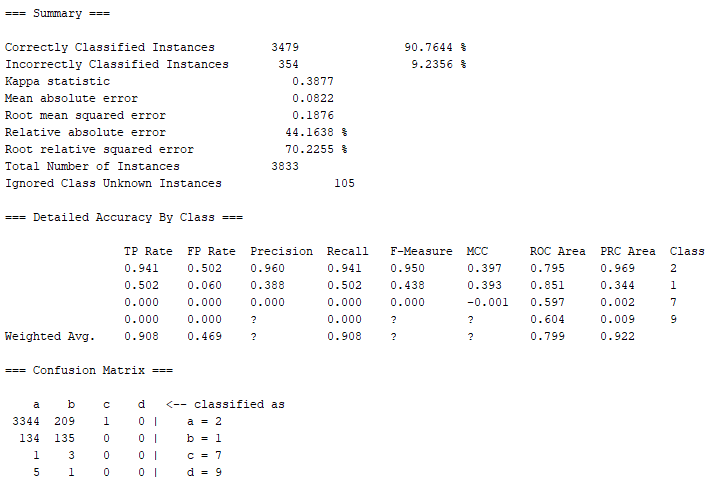
1. **CorrelationAttributeEval**
2. *NaiveBayes*

Accuracy: 90.3209%

**

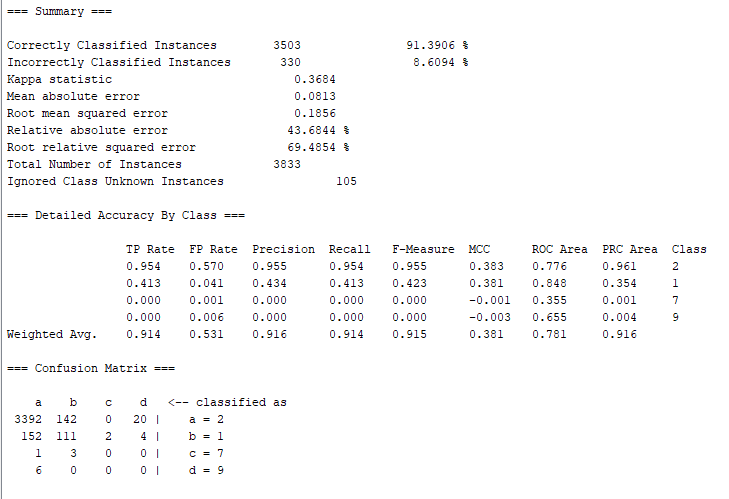
1. *Bagging*

Accuracy: 90.7644%

**

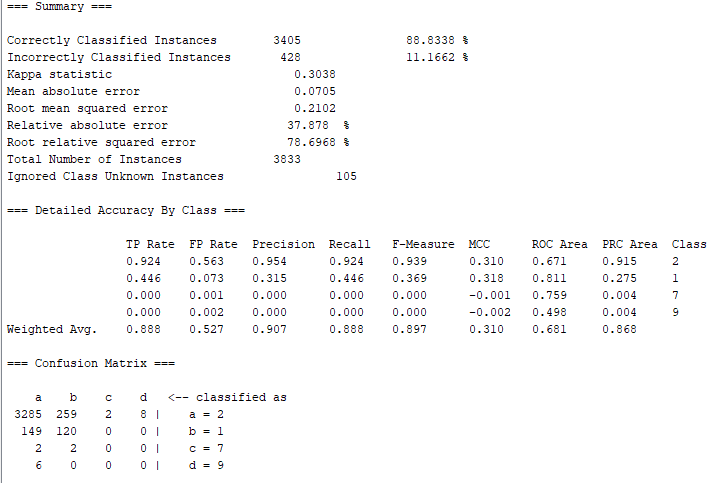
1. *Logistic*

Accuracy: 91.3906%

**

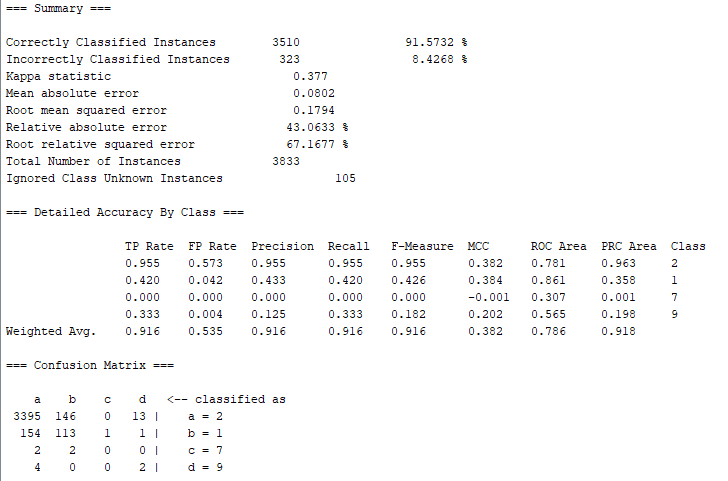
1. *Kstar*

Accuracy: 88.8338%

**

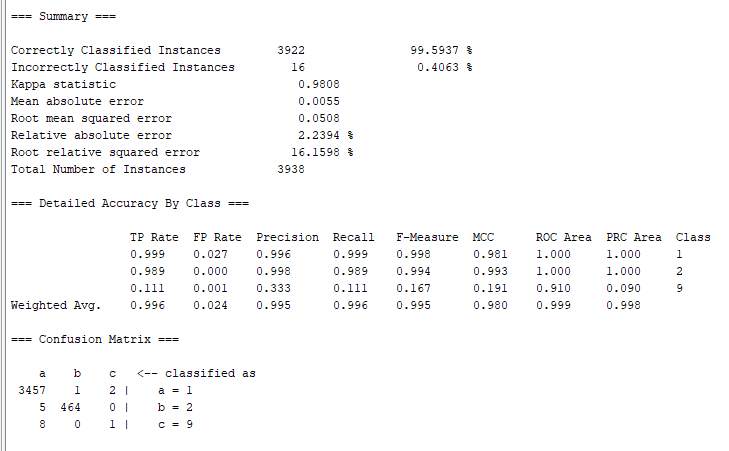
1. *MultipleClassClassifier*

Accuracy: 91.5732%

**

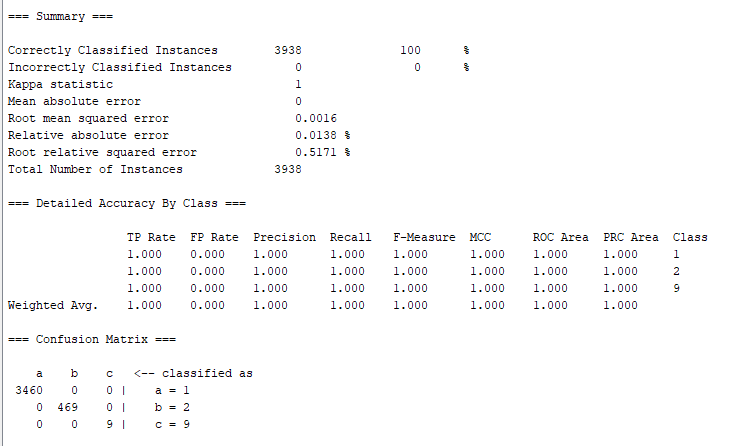
1. **InfoGainAttributeEval**
2. *NaiveBayes*

Accuracy: 99.5937%

**

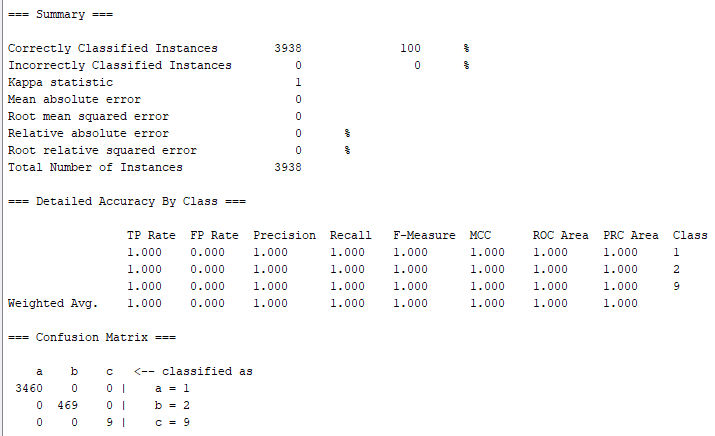
1. *Bagging*

Accuracy: 100%

**

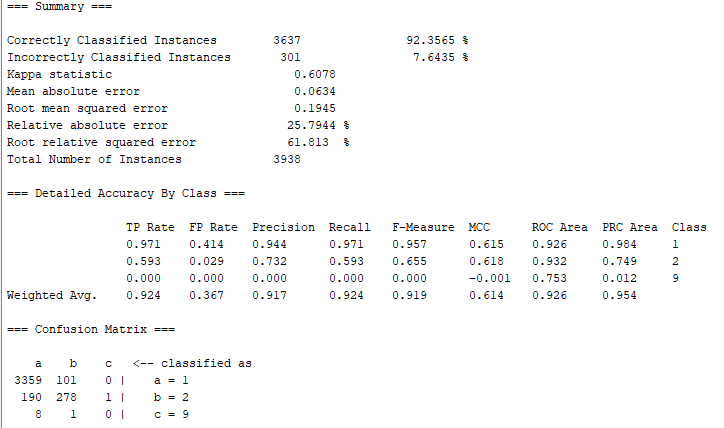
1. *Logistic*

Accuracy: 100%

**

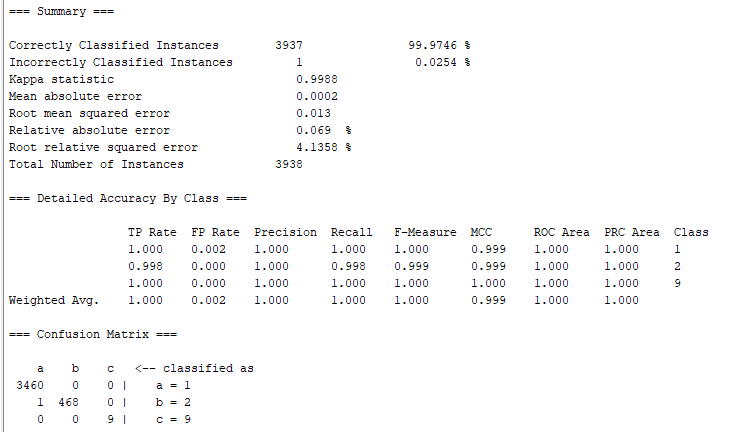
1. *Kstar*

Accuracy: 92.3565%

**

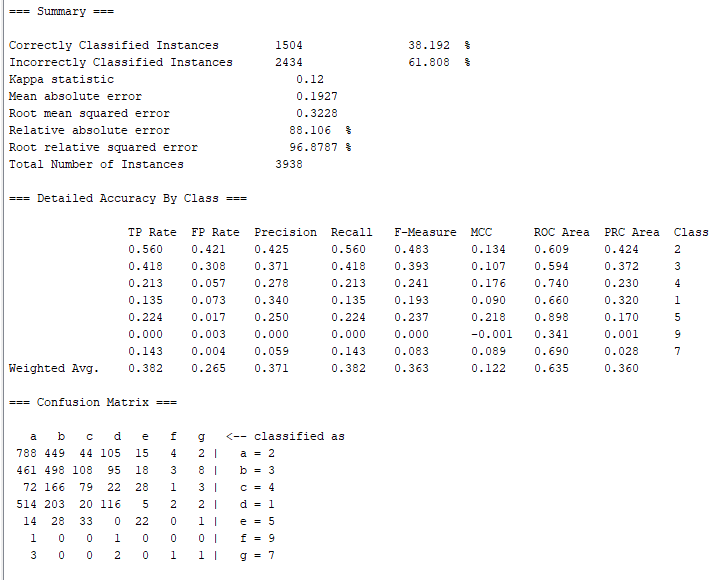
1. *MultipleClassClassifier*

Accuracy: 99.9746%

****

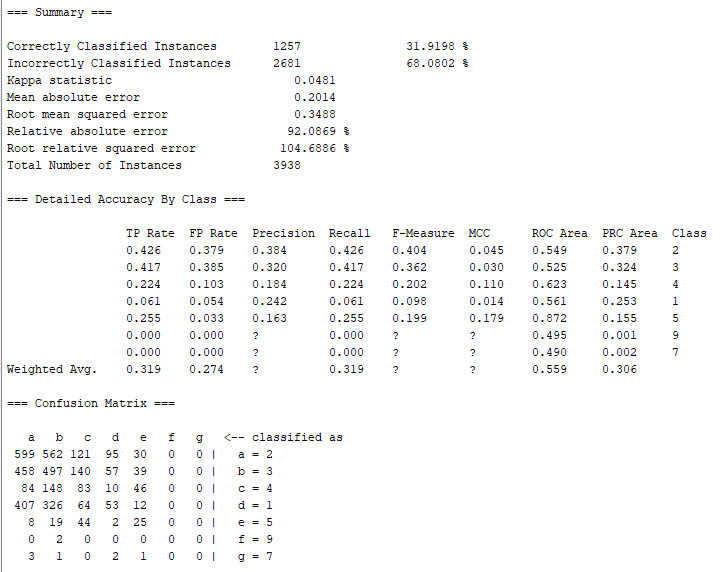
1. **OneRAttributeEval**
2. *NaiveBayes*

Accuracy: 38.192%

**

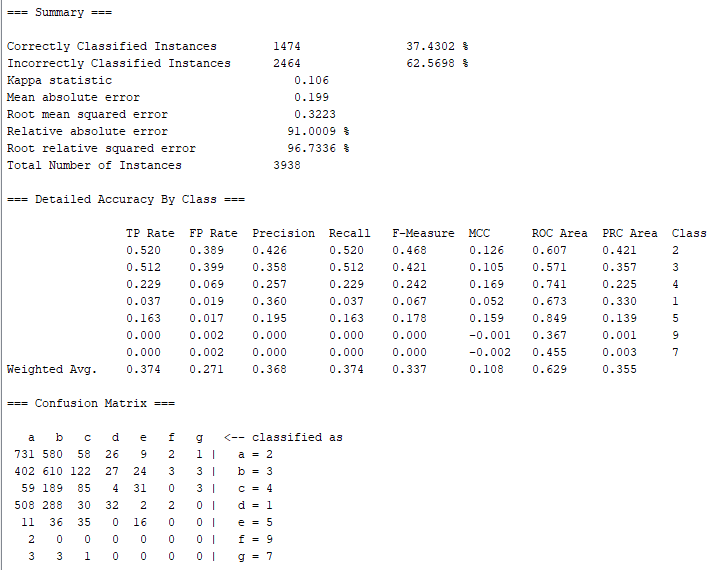
1. *Bagging*

Accuracy: 31.9198%

**

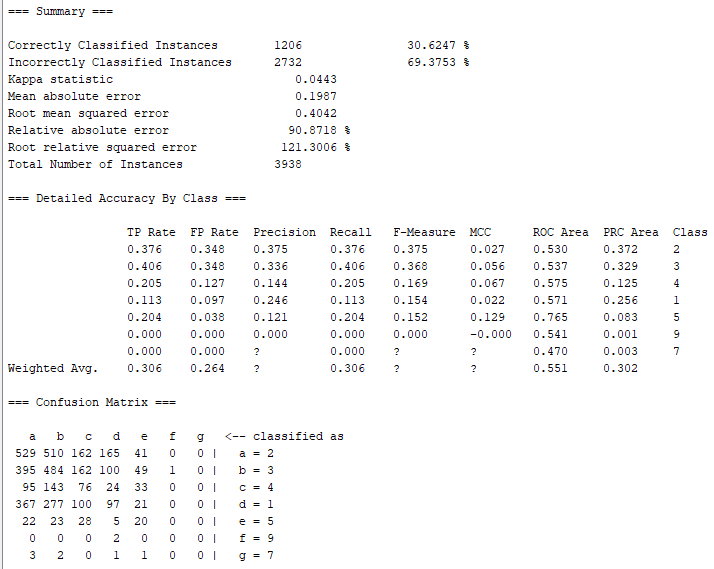
1. *Logistic*

Accuracy: 37.4302%

**

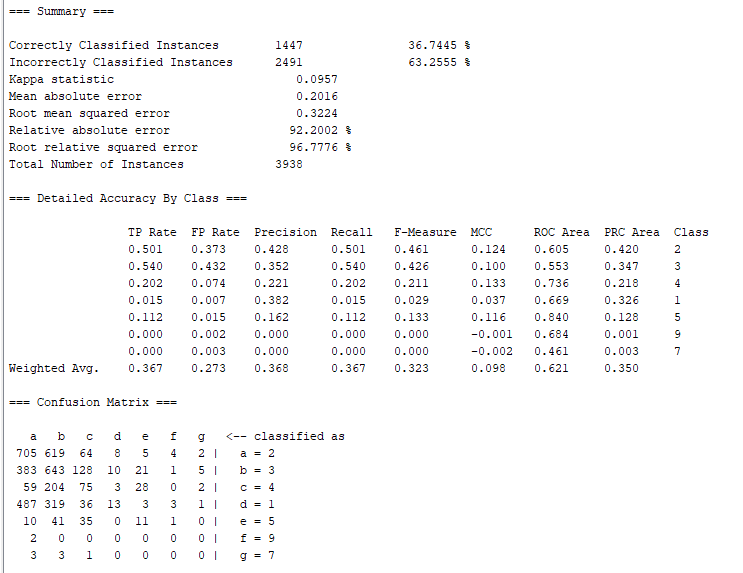
1. *Kstar*

Accuracy: 30.6247%

**

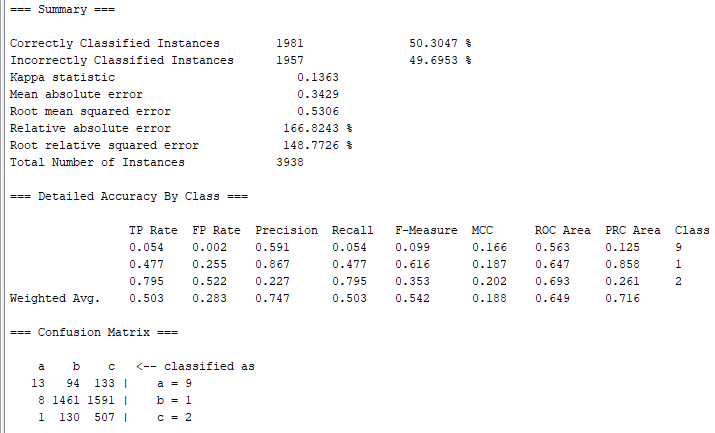
1. *MultipleClassClassifier*

Accuracy: 36.7445%

****

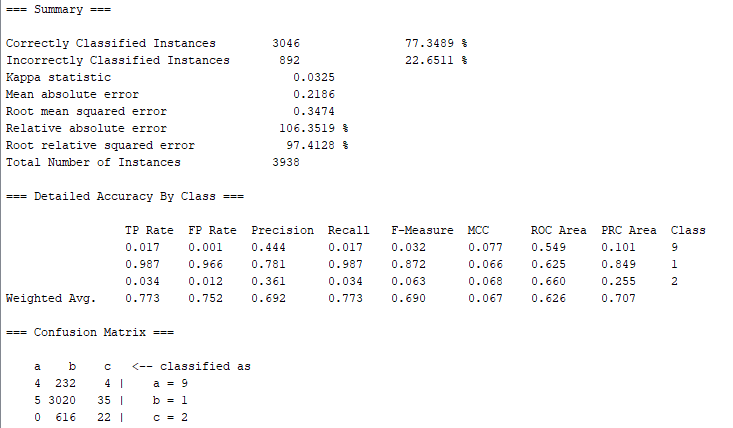
1. **SymmetricalUncertAttributeEval**
2. *NaiveBayes*

Accuracy: 50.3047%

****

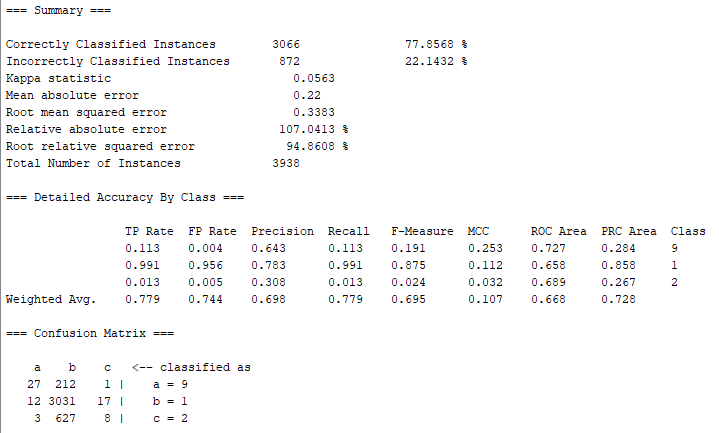
1. *Bagging*

Accuracy: 77.3489%

**

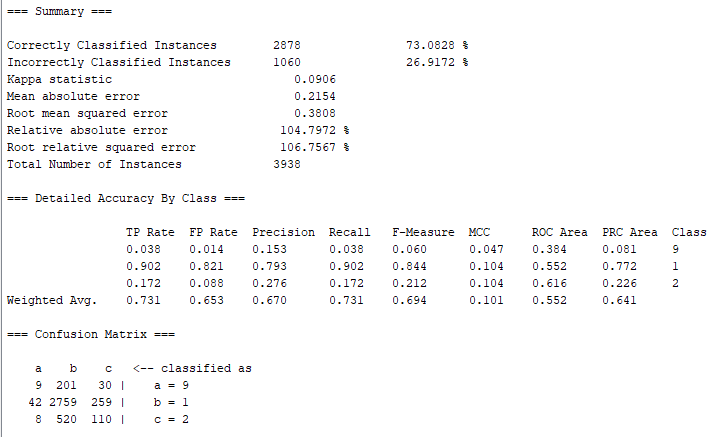
1. *Logistic*

Accuracy: 77.8568%

**

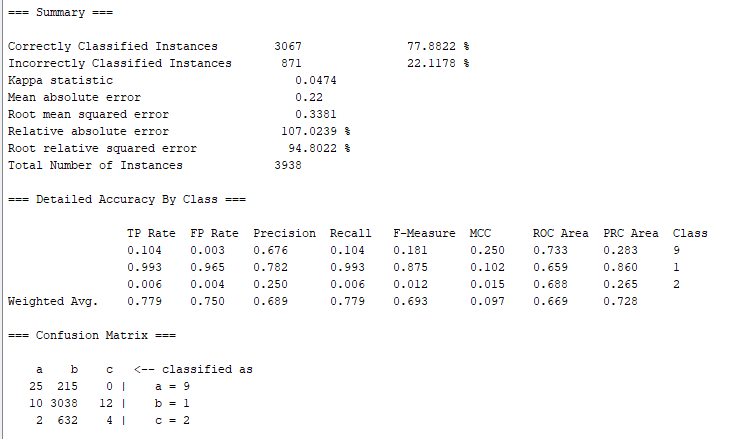
1. *Kstar*

Accuracy: 73.0828%

**

1. *MultipleClassClassifier*

Accuracy: 77.8822%

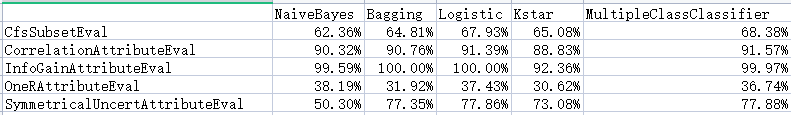
**

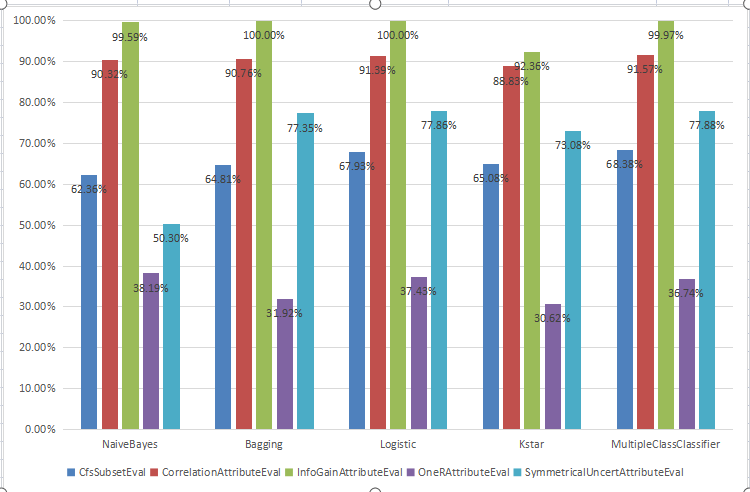
**Better Model**

**Criteria**

1. *Accuracy:*This is the most straightforward criterion. The higher the accuracy, the better the model.
2. *Consistency across different evaluation methods:* A good model should perform well not just in one evaluation method but across different methods. If a model performs well in one method but poorly in another, it might be overfitting to a specific type of data distribution.
3. *Trade-off between simplicity and accuracy:* Simpler models (like NaiveBayes) are easier to understand and interpret, and they also tend to generalize better to new data. So if the accuracy difference is not too big, one might prefer a simpler model.

**Comparation Table**

****

****

**Summary**

Based on the table and the criteria, the *Logistic model* with *InfoGainAttributeEval* method seems to be the best choice as it offers the highest accuracy (100%) and is also relatively consistent across different evaluation methods. However, the final decision should also consider other factors such as the specific application and data characteristics, the computational cost of the model, and the trade-off between model complexity and interpretability.

**Reflections**

***5 attribute I think most relevant***

*x.age80:* This attribute appears in the top five for three out of the five feature selection methods (CorrelationAttributeEval, InfoGainAttributeEval, SymmetricalUncertAttributeEval). This suggests that it is a highly relevant attribute for the class.

*diffwalk:* This attribute appears in the top five for three out of the five feature selection methods (CfsSubsetEval, CorrelationAttributeEval, SymmetricalUncertAttributeEval). This indicates that it is a significant attribute for the class.

*x.ageg5yr:* This attribute appears in the top five for two out of the five feature selection methods (CorrelationAttributeEval, SymmetricalUncertAttributeEval). It also appears in the top 20 for InfoGainAttributeEval, suggesting its relevance.

*employ1:* This attribute appears in the top five for two out of the five feature selection methods (CfsSubsetEval, SymmetricalUncertAttributeEval). It also appears in the top 20 for InfoGainAttributeEval and OneRAttributeEval, indicating its importance.

*checkup1:* This attribute appears in the top five for two out of the five feature selection methods (CorrelationAttributeEval, SymmetricalUncertAttributeEval). It also appears in the top 20 for InfoGainAttributeEval, suggesting its relevance.

These attributes are considered relevant based on their frequency of appearance in the top five across different feature selection methods. However, the specific relevance of these attributes may also depend on the context of your data and the specific class attribute you are predicting.

***I learned from this project***

The project involved the use of various machine learning techniques and algorithms, including Naive Bayes, Bagging, and Logistic Regression. These algorithms were applied in different contexts, demonstrating their strengths and weaknesses. For example, Naive Bayes showed high accuracy in some cases, but lower accuracy in others.

The project also involved feature selection methods like InfoGainAttributeEval and OneRAttributeEval. These methods were used to rank attributes based on their predictive power, which is crucial in improving the performance of the models.

The project demonstrated the importance of preprocessing in a machine learning pipeline, including feature selection and dimensionality reduction.

***Other Observations***

The accuracy of the models varied significantly depending on the feature selection method and the classification algorithm used. This highlights the importance of choosing the right combination of feature selection method and classifier for a given dataset.

The project demonstrated the use of various feature selection methods and classifiers, providing a comprehensive learning experience in applying machine learning techniques.