Practical Application 1

Machine Learning

Yuxiao Xiong

y.xiong@alumnos.upm.es - a18m033

1 Introduction

In this assignment, a Dataset about voice[1] will be applied to the five non-probabilistic supervised classification algorithms (including k-Nearest Neighbors, Rule Induction, Neural Networks, Support Vector Machine and Classification Trees) in order to identify a voice as either male or female. The objective is achieved by using four kinds of analyses (with all original variables, with a univariate filter feature subset selection, with a multivariate filter feature subset selection and with a multivariate Wrapper feature subset selection).

Consequently, we will extract some results and discuss them to analyze the performance of these algorithms if it is possible.

2 Problem Description

The Dataset used in this work consists of 3,168 recorded voice samples, collected from male and female speakers. The information of these samples are represented by 20 numeric acoustic properties and 1 target variable. More specifically, these variables are:

- `meanfreq`: mean frequency (in kHz).
- `sd`: standard deviation of frequency.
- `median`: median frequency (in kHz).
- `Q25`: first quantile (in kHz).
- `Q75`: third quantile (in kHz).
- `IQR`: interquantile range (in kHz).
- `skew`: skewness.
- `kurt`: kurtosis.
- `sp.  ent`: spectral entropy.
- `sfm`: spectral flatness.

- `mode`: mode frequency.
- `centroid`: frequency centroid.
- `peakf`: peak frequency (frequency with highest energy).
- `meanfun`: average of fundamental frequency measured across acoustic signal.
- `minfun`: minimum fundamental frequency measured across acoustic signal.
- `maxfun`: maximum fundamental frequency measured across acoustic signal.
- `meandom`: average of dominant frequency measured across acoustic signal.
- `mindom`: minimum of dominant frequency measured across acoustic signal.
- `maxdom`: maximum of dominant frequency measured across acoustic signal
- `dfrange`: range of dominant frequency measured across acoustic signal.
- `modindx`: adjacent measurements of fundamental frqeuncies divided by the frequency range.
- `label`: target variable, its value is either male or female.

3 Methodology

Before we move further on, it is necessary to pre-process the input data. While the data is very clean and does not contain any empty values, some features have bigger values than others, for instance, `kurt` has a mean value of 36.6 while the mean value of `IQR` is only 0.08. Therefore, we need to normalize numeric variables, using Python, Pandas and Sklearn, to avoid problems in future analysis.

3.1 Algorithms

As we mentioned in the introduction, there are five algorithms that will be implemented.

First of all, the k-nearest neighbor algorithm (**k-NN**) uses proximity to make classifications about the grouping of an individual data point, assuming that nearby points can be founded one another.[2] The value of k is 20 and the hold-one-out cross-validation is active so Weka selects automatically the best k value between 1 and 20. This algorithm is specified as **IBk** in Weka.

Secondly, the **Rule-Induction** algorithm that will be used is called "Repeated Incremental Pruning to Produce Error Reduction (**RIPPER**)". It derives a set of rules from the training set.[3] In Weka, it is showed as **JRip**.

In addition, among all the **Artificial Neural Networks** (NN), we choose the **Multilayer Perceptron**, which is arranged with three layers with input, hidden and output

neurons.[3] In other words, there will be only one hidden layer because the Dataset is 1-dimensional (all its variables describes only one thing: voice). Moreover, the number of neurons in the hidden layer will be $(num\_attributes + num\_classes)/2$.

As for the Support Vector Machine (**SVM**), a non-linear one with a polynomial kernel (exponent=2) is applied because it is more general.[3] This algorithm is showed as **SMO** in Weka.

At last, the **C4.5** algorithm is selected as the Classification Tree. The algorithm choose attributes by maximizing the gain ratio and is known as **J48** in Weka.[3]

While applying all these algorithms, Cross-Validation with 10 folds is used as testing. With this kind of testing, Weka first takes 100 labeled data and produces 10 equal sized sets. Each set is divided into two groups: 90 labeled data are used for training and 10 labeled data are used for testing. Then Weka produces a classifier with an algorithm from 90 labeled data and applies that on the 10 testing data for set 1. Weka repeats these steps for the rest 9 sets and produce 9 more classifies. Finally, Weka averages the performance of the 10 classifiers produced from 10 equal sized sets. [3]

3.2 Feature Subset Selection

Among the four analyses, despite of the first one with all original variables, the rest need to go through the Feature Subset Selection process where irrelevant and redundant variables will be removed as many as possible.

On the one hand, in filtering approaches, there are univariate filters and multivariate filters. The univariate filtering utilizes **Gain Ratio** which evaluates the attributes by measuring the gain ratio respecting to the class.[3] The threshold assigned is 0.1, so the variables with a ratio smaller than 0.1 will not be considered. The multivariate filtering consists of the Correlation-based feature selection (**CFS**) that evaluates by considering the individual predictive ability of each feature along with the degree of redundancy between them. [3]

On the other hand, Wrapper approaches evaluate each possible subset of features with a criterion consisting of the estimated performance of the classifier built with this subset of features.[4] In this case, **Best First** is the method that helps the Wrapper process. The classifiers for each Wrapper approach have the same parameters as the ones used for all variables analysis. Moreover, the evaluation metrics is accuracy for discrete classes and RMSE (Root Mean Square Error)) for numeric classes.

4 Results

After executing five algorithms based on four different subsets generated by the feature selection mentioned in the introduction, we have obtained the following results, where Table 1 shows the results of applying Feature Subset Selection, Table 2 describes the accuracy of every classification algorithm on different subsets, and Table 3 lists the optimistic k value of KNN algorithm chosen by cross-validation for each subset.

	Univariate	Multivariate	Wrapper				
			KNN	RIPPER	SVM	Neural Networks	C4.5
meanfreq			✓				
sd	✓		✓			✓	
median					✓	✓	
Q25	✓				✓	✓	
Q75					✓	✓	
IQR	✓	✓	✓	✓		✓	✓
skew			✓		✓		
kurt							
sp.ent	✓				✓		✓
sfm	✓		✓	✓	✓	✓	✓
mode	✓		✓	✓	✓		
centroid				✓			
meanfun	✓	✓	✓	✓	✓	✓	✓
minfun					✓		
maxfun					✓		
meandom					✓		
mindom					✓		✓
maxdom					✓		
dfrange					✓	✓	
modindx			✓				

Table 1: Attributes selected by univariate and multivariate filters and 5 Wrappers

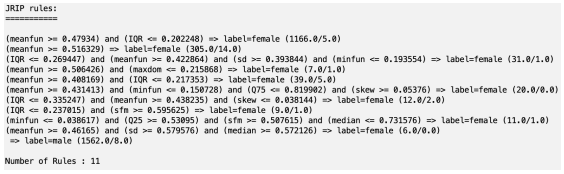
In RIPPER classifiers there are different sets of rules that describe how the algorithm works inside. In this case, these rules are listed in Figure 1.

	KNN	RIPPER	SVM	Neural Networks	C4.5
All variables	98.1061%	96.8750%	97.7273%	97.6641%	96.6856%
Univariate	98.2955%	97.0960%	97.1907%	97.6641%	97.1907%
Multivariate	96.8750%	97.0013%	95.8018%	96.8119%	96.8119%
Wrapper	98.4217%	97.5379%	97.6957%	97.9482%	97.5694%

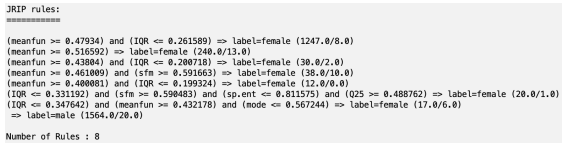
Table 2: Accuracy of the five algorithms on each subset

	k
All variables	3
Univariate	5
Multivariate	11
Wrapper	5

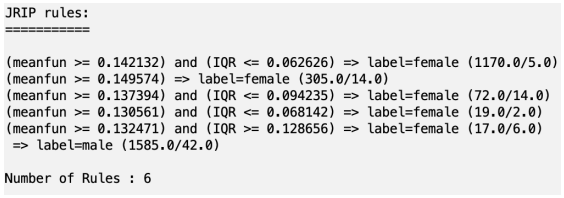
Table 3: Best k value for each subset



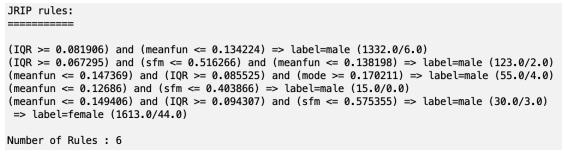
(a) Rules - All variables



(b) Rules - Univariate



(c) Rules - Multivariate



(d) Rules - Wrapper

Figure 1: Rules of RIPPER algorithm

Regarding to SVM, as we can see in Table 4, every execution of the algorithm would produce hundreds of support vectors which make it impossible to investigate the working process inside. Multilayer Perceptron has the same problem as SVM, as showed in Figure 2 where the amount of the links between nodes of distinct layes is huge. In addition, we have decision trees created by C4.5 (Figure 3). The branches indicate every decision been made and end with nodes which specify the final classification. Finally, in Figure 4 there are five confusion matrices that correspond to the best five classifiers in this project.

	Number of support vectors
All variables	252
Univariate	313
Multivariate	597
Wrapper	270

Table 4: Number of support vectors of each SVM classifier

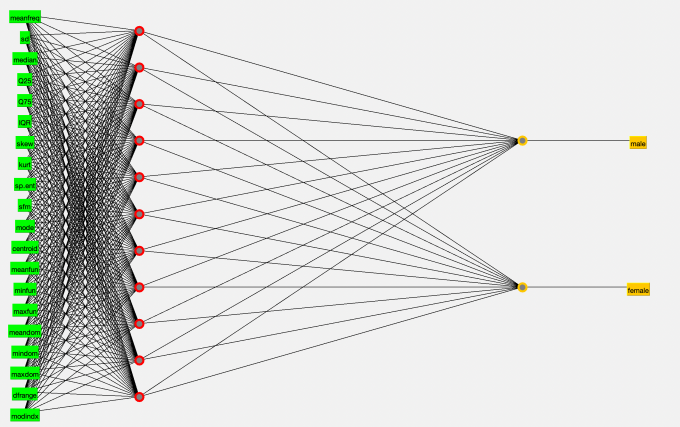


Figure 2: Input, hidden and output layer of Multilayer Perceptron for the original Dataset

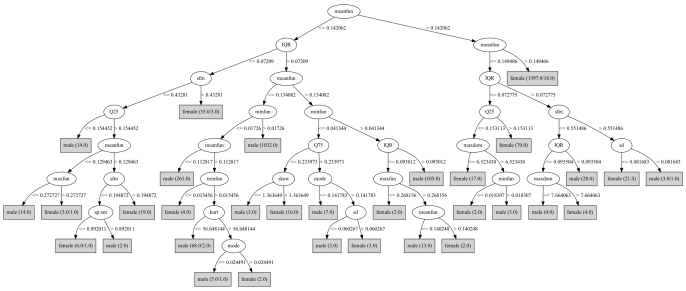
5 Discussion

From Table 2 we can conclude that the classifiers used for the Dataset have an outstanding performance, with an overall accuracy more than 95%. Concretely, two ideas can be deduced:

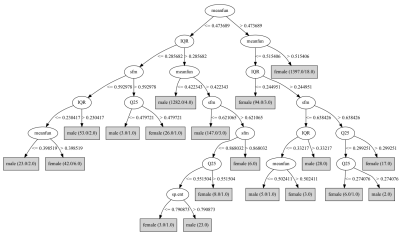
1. The results based on Wrapper approaches are generally better than the ones based on other kinds.
2. In most cases, KNN classifiers produce a better performance on accuracy than other kinds.

5.1 Relevant and Redundant variables

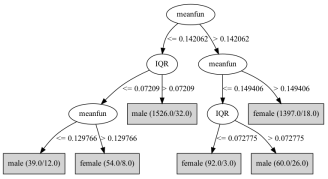
With the help of feature selection, we can identify if a variable is relevant or not to improve the efficiency of the algorithms and accuracy of the classifiers. In this practice, according to Table 1, the variable `meanfun` is the most relevant while `kurt` is the most irrelevant, since `meanfun` is chosen by all filters and Wrappers but no filters or Wrappers select `kurt`. If we observe the histogram of `kurt`(Figure 5), where more than 3,000 of 3,168 instances almost take the same value, we can infer that this variable is meaningless for this



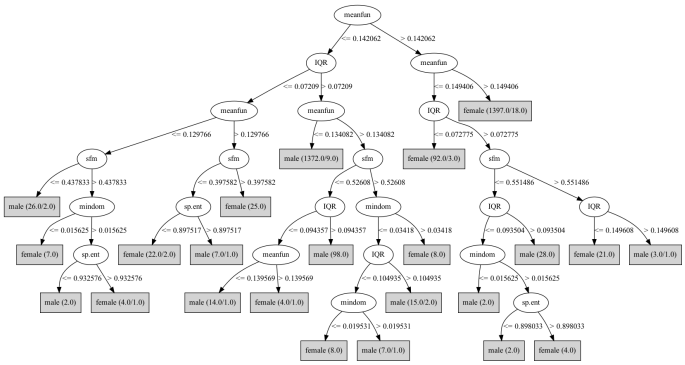
(a) Tree - All variables



(b) Tree - Univariate



(c) Tree - Multivariate



(d) Tree - Wrapper

Figure 3: Decision tree of C4.5

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1556   28 |    a = male
   22 1562 |    b = female
```

(a) KNN - Wrapper

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1554   30 |    a = male
   24 1560 |    b = female
```

(b) KNN - Univariate

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1562   22 |    a = male
   38 1546 |    b = female
```

(c) KNN - All variables

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1545   39 |    a = male
   26 1558 |    b = female
```

(d) NN - Wrapper

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1549   35 |    a = male
   37 1547 |    b = female
```

(e) SVM - All variables

Figure 4: The confusion matrix of the best five classifiers

Dataset.

Besides these two attributes, some variables are also interesting to analyze. For example, `median`, `Q75`, `maxdom`, `meandom` and `dfrange` are only selected by Wrapper-SVM, so they are irrelevant to other classifiers.

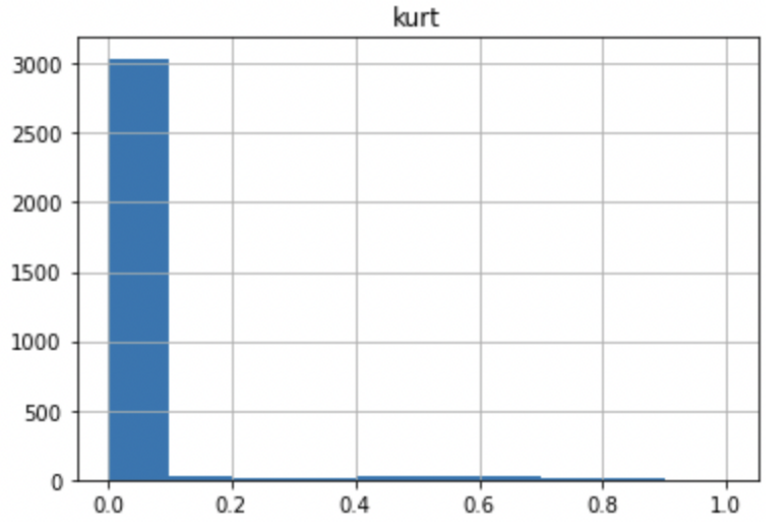


Figure 5: Histogram of the variable `kurt`

5.2 KNN

From Table 2 we can observe that removing irrelevant attributes improves the performance of the classifier, it's obvious that KNN is sensitive to irrelevant variables. Furthermore, KNN performances the best on Wrapper subset and the worst on multivariate subset (only two attributes) in this case, the reason for this could be that the multivariate filter discards too many attributes and the results are less precise.

5.3 RIPPER

Since the rules applied by RIPPER are shown in every execution of the classifier, it is more obvious to know those rules that lead to deduce the target class, such as the rule that implies to the female class according to values of `meanfun` and `IQR`, appears in all feature filters except in the Wrapper one. Comparing the rules and accuracy of the classifiers, we can discover that the multivariate subset has less attributes than the Wrapper one but they produce same number of rules.

In general, RIPPER is not very sensitive to the feature filters because the accuracy differs only a bit in the four subsets.

5.4 SVM

SVM is the only case that the accuracy decreases when discarding attributes. This conclusion is drawn from Table 1 and 2, where the original Dataset has most attributes and highest accuracy while the multivariate selection subset only has two attribute and worst accuracy. Additionally, the accuracy of the Wrapper distinguish only a little from

the one of all variables because the Wrapper approach only removes 6 of 20 variables.

5.5 Multilayer Perceptron

Due to the large amount of links between nodes of distinct layers, this algorithm takes much more time than others, especially when selecting subsets with Wrapper approach. Accuracy of each feature selection classifier is similar.

5.6 C4.5

From Figure 3 we can see the decision trees created by C4.5. An interesting discovery when comparing the trees is that, despite the Wrapper approach choose only 5 of the 20 attributes, some branches of the Wrapper one are more complex than the ones of the all variables. As such, we can deduce that C4.5 is also sensitive to feature filters. Besides, Wrapper-approach classifier leads to better accuracy than the one of the all variables.

5.7 Confusion Matrices

At last, we can explore the confusion matrices produced by the best five classifiers (illustrated in Figure 4), including classifiers of KNN, Neural Networks and SVM. As a binary classification Dataset, although no classifier achieves to classify a class completely correct, the two target classes have confusion numbers in a balanced way.

6 Conclusion

In this practical assignment, we have tried to predict if a voice belongs to a male or a female, based on a Dataset of 20 attributes and around 3,000 observations. We have accomplished the goal by using five non-probabilistic supervised classification algorithms in the case of four analyses about feature subset selection.

Actually, the accuracy already reaches to 95% considering only two attributes: `meanfun` and `IQR`. In general, the winners of the four analyses are KNN and RIPPER, however, in some cases, Neural Networks and SVM have better results than RIPPER. C4.5 has a high and balanced accuracy in each situation but it is not that good as KNN or RIPPER. The reason for this could be the number of leaves of the tree, in other words, we may need a non-binary tree to make the decisions, which can remain as a future work. As for the feature filters, the Wrapper approach wins in most cases.

References

[1] Kory Becker. *Gender Recognition by Voice*. data retrieved from Kaggle, https://www.kaggle.com/datasets/primaryobjects/voicegender. 2016.

[2] *What is the k-nearest neighbors algorithm?* URL: https://www.ibm.com/topics/knn.

[3] Eibe Frank, Mark A. Hall, and Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques*. Online Appendix - The Weka Workbench. 2016. URL: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.

[4] Pedro Larrañaga and Concha Bielza. *Feature Subset Selection*. 2022.