Classification and Analysis of Mushroom Dataset

Fernando Lahiru

Abstract- This report will aim to look at the illustration and documentation of how the classification of the mushroom dataset can be used as an aid to differentiate poisonous and edible mushrooms, as well as analysing the different classifiers.

Introduction

Due to the evolving of computer science and the fast development and vast usage of World Wide Web and other electronic data, information extraction is a popular research field. Data mining is the process of analysing data from different perspective and summarizing it into useful information. The main aim of data mining is to uncover relationship in data and predict the outcome [1]. Data mining involves many methods such as clustering, classification, summarization and sequential pattern matching. The three different steps of learning procedure are presented in the diagram below.

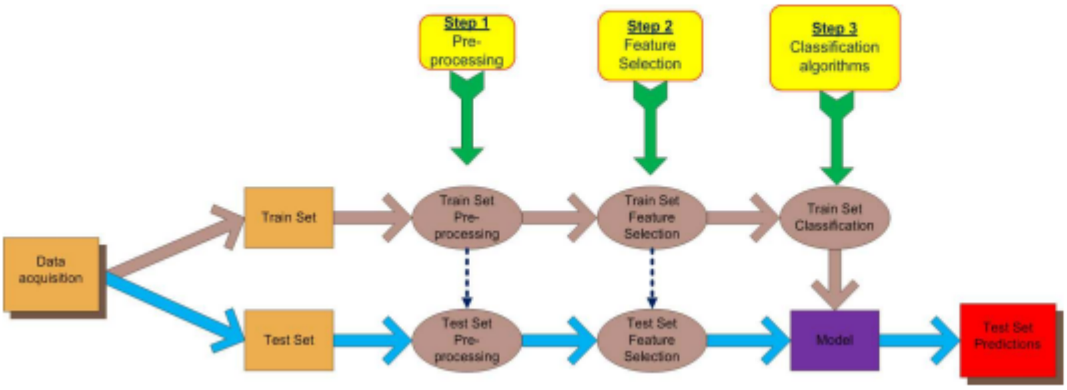


Figure 1 Flowchart of learning procedure

Classification is one of the most important techniques of data mining. Classification is the process of finding a set of models (or functions) which describe and distinguish data classes or concepts. [2]

In classification, inputs are given a set of data, called a training set, where each record consists of several fields or attributes. One of these attributes, called the classifying attribute, indicates the class to which each dataset belongs. The object of classification is to build a model of the classifying attribute based upon the other attributes which are not from the training dataset. [3]

For the implementation of classification, other techniques need to also be applied such as pre-processing. Before data mining algorithms can be applied, a target dataset must be created. This is because real-world data can be incomplete (i.e. lacking attribute values), noisy (i.e. containing errors) and errors. This means that they are not ready to be considered for the data mining stage. Therefore, pre-processing is one of the most critical steps, which needs to be performed in order to prepare and transform the dataset. Data pre-processing includes data cleaning, normalization, transformation, feature extraction and selection etc…The product of data pre-processing is the final training set [4].

Data cleaning is a task in pre-processing, which deals with detecting and removing errors and inconsistencies from data in order to improve the quality of the data [5]. Missing values is one common problem that data cleaning tackles.

There are different types of classification. These include: decision trees, logistic regression and Naive Bayes. Decision tree is a classification algorithm, which repeatedly splits the data set according to a criterion that maximizes the separation of the data, resulting in a tree-like structure. The most common criterion employed is information gain; this means that at each split, the decrease in entropy due to this split is maximized. They have the advantage that they can easily be expressed as rules. A disadvantage of decision trees is that the combination of single best variable and optimal split-point is selected, however a multi-step lookahead that considers combinations of variables may obtain different results [6].

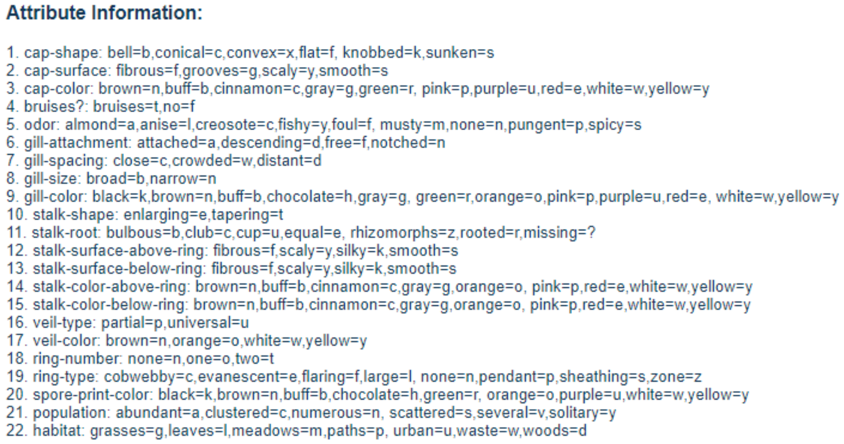
Logistic regression is a classification method, which "models the probability of occurrence of one (success) of the two classes of a dichotomous criterion". Dichotomous means there are only two possible classes. This is one of the most popular machine learning algorithms for binary classification. In order to make predictions with a logistic regression model, you just have to plug in values into the logistic regression equation and calculate a result. This algorithm is simple and easy to implement; however, it is not able to handle a large number of categorical features/variables.

Naïve Bayes classifier is the simplest of Bayesian models, in that it assumes that all attributes of the examples are independent of each other given the context of the class. [7] It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It is easy to build and particularly useful for large data sets. However, if a categorical variable has a category, which was not observed in the training dataset, then the model will assign it a zero probability, and will be unable to make a prediction.

59.8% literature on data analysis indicates that decision tree is the most preferred algorithm for data analysis. [8] Therefore, the motivation behind this study to be able to choose the most effective machine learning classification algorithm for this kind of data.

Dataset

In this report, for the analysis and classification, mushroom dataset is considered. This database contains 22 attributes and 8124 instances. It is 374KB and is freely available on the internet, on the UCI Machine Learning Repository. This dataset contains descriptions of hypothetical samples corresponding to 23 of gilled mushrooms. The "class" field refers to the edibility of the mushroom, i.e. whether the species is identified as edible or poisonous. This is marked as "p" for poisonous or "e" for edible. The figure below shows the other attributes present in this dataset:



Attribute Information:

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

Figure 2: Attributes Information

Background

There are 22 attributes to be considered in this dataset and each one of them describes a particular aspect of the mushroom. A mushroom is the construction of a fungus, normally situated above the ground on soil scavenging on nutrients in research of food source.

To identify a mushroom can be a quite tricky situation as you cannot tell as first close if one is edible or not or poisonous from non-poisonous. As well as their shape and colour would also help to identify their nature. The spore of a mushroom named basidiospores [9],produced on the gills allows them to produce a rain of powder under the cap. In a mushroom there are different attributes that needs to be considered in order the structure of such.

The cap is the skeleton which is based at the top of the mushroom meaning is correlates to the head of the mushroom. The cup(volva), base of the mushroom is the pre-existing base of the mushroom before developing into a mature mushroom. This can be found in some, not all have it. Gills are arranged at the centre located at the lower-side of the mushroom cup. The spores are born and produced from the gills.

The ring of the mushroom is the remaining of the velvet (this is the tissue that allows the connection of stem and the cap before the mushroom develops into its new body. Some mushroom would have a feature that can only be seen and analysed when cut or either bruised this is when their original colour would drastically change pigment, by doing so it can be analysed and be able to be classified depending on its colour and be identified. A feature to be considered when describing a mushroom is their veil, a mushroom has 2 different types of veil: a universal veil and a partial veil.

The universal veil is the premature tissue that constructs the mushroom before becoming a matured and fully-grown mushroom. The process of such would be due and the tissue would rupture and be slowly dissolve, by doing so the mushroom would expand and mature gradually, however as we humans have a scar from our birth which is the scar from the incision of the belly button a mushroom would a similar aspect and would have traces of the veil within itself.

The partial veil or inner veil consists of a temporary tissue found in some mushrooms. The role of such is to protect the body of the fungus while its producing the section which will later be used to produce the spores, this is found on the lower surface of the mushroom cap.

Methodology

Pre-processing

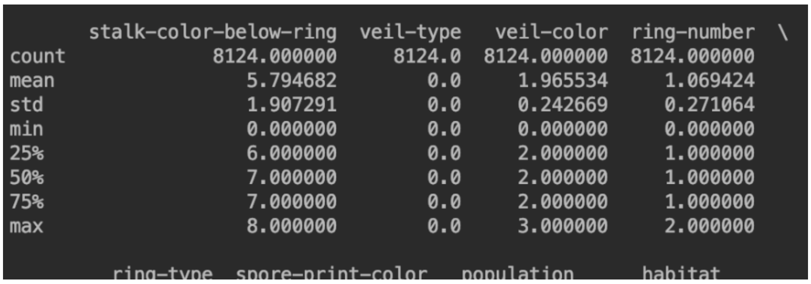
At a first glance, the first thing that was noticed was that the mushroom dataset contains missing values (e.g. "?"). Hence, data cleaning was needed in order to handle missing values before the data mining stage. There are several techniques that could be done to handle this.

One technique for handling missing data is ignoring the tuple, which means that you ignore the rows which contain a missing value. This is the simplest strategy for handling the missing data, however can be seen as ineffective because it reduces the sample size.

Moreover, a second technique that could be used to handle missing values is deleting columns which have more than a certain percentage of missing values. This is not very effective because it means that you are losing the whole column because of some missing values.

Another technique is to fill the missing value manually. This is the most common approach, whereby missing values are replaced with a test statistic e.g. mean, median or mode of the particular feature the missing values belong to. However, it affects the variability of the data and using mean is greatly affected by outliers in the data, which in turn could affect the effectiveness of the data.

Therefore, the technique which I used for my study was the removal of columns, which contain more than 20% of missing values. This resulted in the "veil-type" column having more than 20% of missing values, therefore this column was omitted. This can be shown in the figure below:



```
      stalk-color-below-ring  veil-type   veil-color  ring-number  \
count            8124.000000     8124.0  8124.000000  8124.000000
mean                5.794682        0.0     1.965534     1.069424
std                 1.907291        0.0     0.242669     0.271064
min                 0.000000        0.0     0.000000     0.000000
25%                 6.000000        0.0     2.000000     1.000000
50%                 7.000000        0.0     2.000000     1.000000
75%                 7.000000        0.0     2.000000     1.000000
max                 8.000000        0.0     3.000000     2.000000

         ring-type  spore-print-color   population       habitat
```

Figure 3: "Veil-type" column

Next, data transformation was required because all the data was categorical, which meant that this data could not be fed into classifiers. Hence, LabelEncoder was used to convert categorical values into ordinal values.

A final check, that must be done before running a classifier, is to see whether the dataset is balanced or not. Imbalanced data is where the classes are not represented equally. This can provide misleading classification accuracy. Therefore, I plotted a graph of the "label" column, which is the classifying attribute in this dataset, in order to check if it is balanced. This can be seen in the figure below.

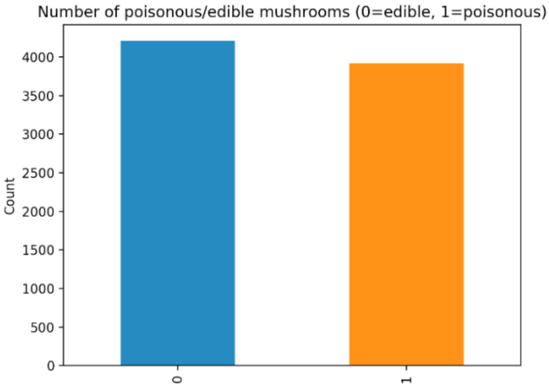


Figure 4: Label Balance

This graph shows that the number of poisonous and edible mushrooms is roughly the same, which means that the dataset is balanced. Therefore, no resampling strategy is required for this dataset.

Classification

In the learning step, the classification model built the classifier by analysing the training set. In the classification step, the "labels" for the data were predicted. The dataset tuples and their associated labels were then split into training set and testing set using sklearn and 3 classifier models were built: decision tree, logistic regression and naïve Bayes. The test set was used to estimate the predictive accuracy of a classifier.

The decision tree was implemented using the sklearn library. This allowed the input of 2 variables, an array X correlating to the number of sample and features, and a Y integer, which includes n_samples.

The logistic regression can handle dense and sparse unit, meaning that is optimal for 64-bit floats, which increases performance. The idea of logistic regression is to "find a relationship between features and probability of particular outcome". As the decision tree takes into input 2 variables which are X and Y, in this case it would returns a confidence score x sample class, therefore predicting the class.

The Naïve Bayes classifier takes into consideration 2 parameters which are X and Y, in this case the sklearn library helps to create such model via utilising the Gaussian model, which is used for the likeness of the features. As shown below in the equation, the 2 parameters have been used to estimate using maximum likelihood.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

Figure 5:Gaussian Naive Bayes Equation

Classification accuracy is calculated on all classifiers on this dataset, which is the "number of correct predictions made as a ratio of all predictions made". However, just classification accuracy is not enough in order to compare the performance of the models. Therefore, a classification report is created, which contains precision, recall and F1 score. Precision is "the number of positive predictions divided by the total number of positive class values predicted". This equation is represented below:

$$\frac{no.of\ true\ positives}{no.of\ true\ positives + no.of\ false\ positives}$$

False positive defines the number of detected attacks which are actually normal. False negative means wrong prediction i.e. it means it detects instances a normal but in actual they are attacks. True positive means instances that are correctly predicted as normal. True negative means instances that are correctly classified or detected as attack. [10]

Recall is the ratio of correctively predicted positive observations to the total predicted positive observations in the test data. It is also called "Sensitivity" or the "True Positive Rate". The equation for this is represented below:

$$\frac{no.of\ true\ positives}{no.of\ true\ positives + no.of\ false\ negatives}$$

F1 score is weighted average of precision and recall. Therefore, this score "takes both false positives and false negatives into account". It is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

$$\frac{2(recall * precision)}{(recall + precision)}$$

Results

In this section, I will be discussing about what I have discovered from running the methodology. The classification performance of the decision tree, logistic regression and Naïve Bayes on the Mushroom dataset are judged on the basis of parameters such as accuracy, precision, recall and F1 Score. Table 1 shows the values of these parameters.

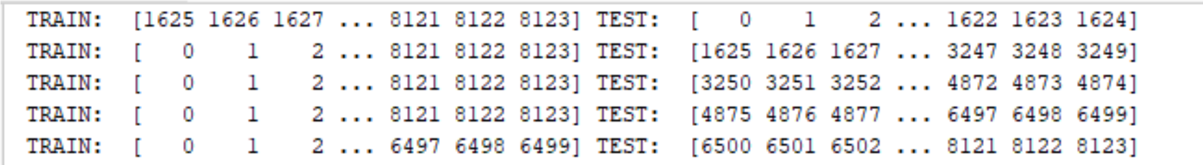
Table 1 Comparison of classification methods

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Decision Tree	100	100	100	100
Logistic Regression	96.187	96	96	96
Naïve Bayes	91.390	91	91	91

Thorough analysis of the above table reveals clear of the three algorithms' effectiveness to the Mushroom dataset. For comparative explicit evaluation, I have compared the algorithms only with respect to the 4 most commonly used classification evaluation measures of accuracy, precision, recall and F1 score. The table shows the accuracy for decision tree, logistic regression, naïve Bayes are 100%, 96.2% and 91.4% respectively. The comparative analysis concludes that Decision Tree outperforms the other classification algorithms.

However, one problem with the current methodology that I noticed was the way training set and testing set were split. It was split by 90/10, which means that there is 90% training data and 10% of testing data. This can cause overfitting, which means that the model was trained "too well" and is now fit too closely to the training dataset. Therefore, it would be accurate on the training data, but might not be very accurate on untrained/new data.

Hence, in order to avoid this, cross validation was needed. The aim of this is to find a balance between building a model that classify the training data effectively without overfitting to the random fluctuations in the training data. In my study, this means that instead of doing the 90/10 train/test split, cross validation can be performed, which applies to more subsets. Therefore, I chose to perform K-Folds cross validation on my code in order to improve the validity of my results. The procedure has a single parameter called K, which refers to the number of folds that the given data is to be split into. K – 1 of the folds is used to train the data, and the last fold is used as testing data. This can be shown in the figure below.



```
TRAIN:  [1625 1626 1627 ... 8121 8122 8123] TEST:  [   0    1    2 ... 1622 1623 1624]
TRAIN:  [   0    1    2 ... 8121 8122 8123] TEST:  [1625 1626 1627 ... 3247 3248 3249]
TRAIN:  [   0    1    2 ... 8121 8122 8123] TEST:  [3250 3251 3252 ... 4872 4873 4874]
TRAIN:  [   0    1    2 ... 8121 8122 8123] TEST:  [4875 4876 4877 ... 6497 6498 6499]
TRAIN:  [   0    1    2 ... 6497 6498 6499] TEST:  [6500 6501 6502 ... 8121 8122 8123]
```

Figure 6 Train/Test Split using K-Fold

This shows the specific observations chosen for each train and test set. With this K-Fold, accuracy score across each fold was calculated, in the same way as it was calculated before, and average across all folds was computed. This can be seen in the table below.

Model	**Fold 1: Accuracy Score (%)**	**Fold 2: Accuracy Score (%)**	**Fold 3: Accuracy Score (%)**	**Fold 4: Accuracy Score (%)**	**Fold 5: Accuracy Score (%)**	**Average Accuracy Score (%)**
Decision Tree	100	100	100	99.38	96.55	99.01
Logistic Regression	67.57	96.66	94.65	82.15	97.17	86.44
Naïve Bayes	47.2	52.8	88.68	77.41	95.07	72.22

The table above shows the accuracy score of each classifying, after undergoing cross validation. During the process, the score of this validation allows you to assess the predictive performance, and to view how well the performance worked outside the sample of data versus the new data that is trained. The previous table showed an accuracy of 91-100% between the three classifiers, as it did not use any kind of cross validation and just used one sample set. The new table shows a huge difference with the accuracy score varying from as low as 47.2% for Naïve Bayes and as high as 100%. This proves that, in the previous table, there was clearly overfitting in the data and thus, made the accuracy score inaccurate.

An advantage of using cross-validation is that when we fit a model, we are fitting it to a training set, on the other hand, without cross-validation, there would only be information related to how well the model performs on the sample data. Additionally, using this validation, as shown in the table, allows you to understand the confidence given to the model when trained in the training set.

Conclusion

In this research, the three classification techniques such as decision tree, logistic regression and Naïve Bayes were used to evaluate the percentage of accuracy, precision, recall and F1 score for the mushroom dataset. Several modifications were done to the dataset before undergoing classification i.e. data cleaning and data transformation. Running the three classification models has shown that decision tree, as hypothesized, is the most efficient algorithm, compared to the others.

However, this did not mean that the results were valid because classification models were done on a specific subset of the data. Therefore, cross-validation was performed. Cross validation allowed me to analyse the machine learning models with a better confidence level. in the first case, we have an overfitting problem, which affected the accuracy of the models. Therefore, cross-validation improved the validity of the results which I have obtained.

As a possible future development, this present work can be extended in several directions, one such direction could be re-exploring these algorithms with different datasets and seeing how these compare with my current results or exploring different algorithms with the same datasets.

Limitations

Because of the missing values, extensive data cleaning was required, which reduced the number of attributes, reducing the accuracy. However, had a different dataset been picked without missing values, this could have improved the accuracy of the classification models.

Another limitation is that there are no previous data on the edibility of mushrooms, which means that there is no comparison for this dataset, which could have been used in order to predict whether mushrooms are edible or poisonous.

References

[1] R.P.Lippmann, "Pattern classification using neural networks," IEEE Communication, 1989. [Online].

[2] J. H. M. Kamber, Data Mining: concepts and Techniques, ELSEVIER, 2006.

[3] R. A. M. M. John Shafer, "SPRINT: A scalable parallel classifier for data mining," IBM Almaden Research Center.

[4] D. K. a. P. E. P. S. B. Kotsiantis, "Data Preprocessing for Supervised Leaning," 2006. [Online]

[5] E. R. a. H. H. Do, "Data Cleaning: Problems and Current Approaches," 2000. [Online].

[6] S. D. L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," October 2002. [Online]

[7] I. Rish, "A Comparison of Event Models for Naive Bayes Text Classification," 1998. [Online].

[8] N. N. S. Purohit, "Comparative Study of Binary Classification Methods to Analyze a Massive Dataset on Virtual Machine," 2017. [Online].

[9] G. Garziano, "Mushroom Classification - Part 1," February 2018. [Online].

[10] T. G. S. S. Khurana, "Comparison of Classification Techniques for Intrusion Detection Dataset Using WEK," 2014. [Online].