Mushroom Classification

Ana Marojević[1], Božo Bjeković[2], Maja Miljić[3]
University of Novi Sad/Faculty of Technical Sciences/Software Engineering and Information Technologies,
Novi Sad, Republic of Serbia[1,2,3]
ana94ns@hotmail.com[1], bozo.bjekovic@gmail.com[2], majamiljic2@gmail.com[3]

***Abstract*—This paper focuses on the use of classification techniques for analyzing mushroom data set. Data set is composed of records of different types of mushrooms, which are edible or poisonous. Three different classification algorithms like Support Vector Machines, Naïve Bayes and K-Nearest Neighbors are used to categorize different mushrooms. The performance is evaluated using F1 score. After analyzing it was found that SVM and KNN gave best results with the highest score. On the other hand, Naïve Bayes gave a bit lower performaces, but it is still acceptable.**

Keywords—Machine Learning; Algorithm; Naïve Bayes; k-Nearest Neighbor; Support Vector Machines; kNN; SVM; data

I. INTRODUCTION

This paper presents the use of different classification techniques on mushroom data to classify various types of mushrooms as edible or non-edible. For better understanding classificaton techniques, first, we will introduce You to machine learning concepts.

Machine learning is a type of artificial intelligence (AI) that provides computers ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data. [1]

The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension -- as is the case in data mining applications -- machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervized. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets.

Classification is the process of finding a model or a function that describes and distinguishes data classes and concepts. Classification enables us to predict the classes of objects whose class label is not known. Classification is done using two steps learning and testing. [2]

At the beginning there will be information about data set-short description, where records are drawn from, how many instances and how many attributes there are.

In Chapter III we will write about algorithms in general. There will be shown a division into two groups of algorithms and short desription for every one of them.

Next, since we used three different algorithms to solve our problem, it will be said more about them in Chapter IV. (Titles: *Gaussian Naïve Bayes*, *Support Vector Machines* and *K-Nearest Neighbor*)

More over, we will make comparison between different algorithms and show the statistics for them (Chapter V).

At the end, we will drow a conclusion and write about it in the Chapter VI.

II. DATA SET

The data set we used includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom.

Mushroom records drawn from The Audubon Society Field Guide to North American Mushroom (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf. [3]

Number of Instances: 8124 and number of attributes: 22.

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

III. ALGORITHMS

There are so many algorithms available, but we will divide them in two groups to make it easier.

- The first is a grouping of algorithms by the **learning style**
- The second is a grouping of algorithms by **similarity** in form or function

Both approaches are useful, so in this paper it will be described both. [4]

1. Algorithms grouped by learning style

There are different ways an algorithm can model a problem based on its interaction with the experience or environment.

It is popular in machine learning and artificial intelligence textbooks to first consider the learning styles that an algorithm can adopt.

There are only a few main learning styles or learning models that an algorithm can have and we'll go through them here with a few examples of algorithms and problem types that they suit.

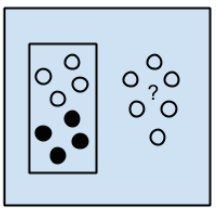
This taxonomy or way of organizing machine learning algorithms is useful because it forces you to think about the roles of the input data and the model preparation process and select one that is the most appropriate for your problem in order to get the best result.

There are three different learning styles in machine learning algorithms:

- *Supervised learning*

Input data is called training data and has a known label or result such as spam/not-spam or a stock price at a time.

A model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data.



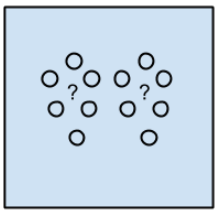
Supervised Learning Algorithms

Exapmle prolems are classification and regression.

- *Unsupervised learning*

Input data is not labeled and does not have a known result.

A model is prepared by deducting structures present in the input data. This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarity.



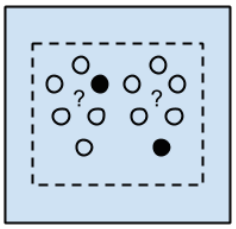
Unsupervised Learning Algorithms

Example problems are clustering, dimensionality reduction and association rule learning and example algorithm is k-Means algorithm.

- *Semi-supervised learning*

Input data is mixture of labeled and unlabeled examples.

There is a desired prediction problem but the model must learn the structures to organize the data as well as make predictions.



Semi-supervised Learning Algorithms

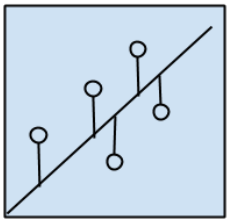
Example problems are classification and regression and example algorithms are extensions to other flexible methods that make assumptions about how to model the unlabeled data.

2. Algorithms grouped by similarity

In this section we will mention a couple groups of algorithms which we think are important.

- *Regression algorithm*

Regression is concerned with modeling the relationship between variables that is iteratively refined using a mesure of error in the predictions made by the model.



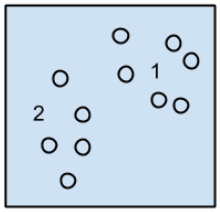
Regression Algorithms

The most popular regression algorithms are Ordinary least squares regression and Linear regression

- *Clustering Algorithms*

Clustering, like regression, describes the class of problem and the class of methods.

Clustering methods are typically organized by the modeling approaches such as centroid-based and hierarchal. All methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonality.



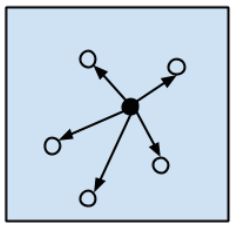
Clustering Algorithms

The most popular clustering algorithm is k-Means.

- *Instance-based algorithm*

Instance-based learning model is a decision problem with instances or examples of training data that are deemed important or required to the model.

Such methods typically build up a database of example data and compare new data to the database using a similarity measure in order to find the best match and make a prediction. For this reason, instance-based methods are also called winner-take-all methods and memory-based learning. Focus is put on the representation of the stored instances and similarity measures used between instances.

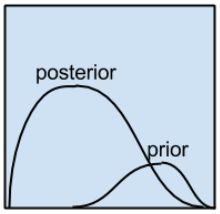


Instance-based Algorithms

The most popular instance-based algorithm is k-Nearest Neighbor (kNN).

- *Bayesian Algorithms*

Bayesian methods are those that exlicitly apply Bayes' Theorem for problems such as classification and regression.



Bayesian Algorithms

The most popular Bayesian algorithms are Naïve Bayes, Gaussian Naïve Bayes, Multinomial Naïve Bayes, etc.

IV. IMPLEMENTATION

After reading the csv file, the file is divided into X and Y variables. Then, the data set is divided into two before doing classification: training set and test set. Since attribute characteristics are categorical, label encoder is used to transform them into numerical values.

The next section provides information about algorithms used for solving this classification problem.

A. Gaussian Naïve Bayes

Naïve Bayes classifier is one of the most effective machine learning algorithms implemented in machine learning projects. Primarily Naïve Bayes is a linear classifier, which is a supervised machine learning method and works as a probabilistic classifier as well. Most of the time, for the numeric implementations K-Nearest Neighbors and K-Means clustering algorithms can be implemented. Naïve Bayes classifier works effectively for classifying emails, texts, symbols, and names. It's not unusual Naïve Bayes classifier is used for numeric data as well in some instances. Naïve Bayes classifier can be implemented on high-dimensional datasets effectively as well. Naïve Bayes classifier predicts the probability of each class based on the feature vector for text classification for continuous big data with a prior distribution of the probability, tackling the challenges of the curse of the dimensionality. There are three types of Naïve Bayes classifiers. When handling real-time data with continuous distribution, Naïve Bayes classifier considers that the big data is generated through a Gaussian process with normal distribution. Multinomial Naïve Bayes classifier can be applied when handling event models where the events are modeled through a multinomial distribution. In this situation, the features are frequencies. In the third scenario, when the features are Boolean or independent, the features are generated through a Bernoullian process. In this scenario, a Bernoulli Naïve Bayes classifier can be applied.

When dealing with Gaussian Naïve Bayes classifier, the outcome model will have a high-performance with high training speed with the capabilities to predict the probability of the feature that belongs to certain class. [5]

B. Support Vector Machines

Support Vector Machines belong to the class of Kernel Methods and are rooted in the statistical learning theory. As all kernel-based learning algorithms they are composed of a general purpose learning machine (in the case of SVM a linear machine) and a problem specific kernel function. Since the linear machine can only classify the data in a linear separable feature space, the role of the kernel-function is to induce such a feature space by implicitly mapping the training data into a higher dimensional space where the data is linear separable. Since both, the general purpose learning

machine and the kernel function can be used in a modular way, it is possible to construct different learning machines characterized by different nonlinear decision surfaces. As mentioned before, the classifier of a Support Vector Machine can be used in a modular manner (as the kernel function) and therefore, depending on the purpose, domain, and the separability of the feature space different learners are used. There is for example the Maximum Margin Classifier for a linear separable data, the Soft Margin Classifier which allows some noise in the training data or Linear Programming Support Vector Machines for classification purposes, but also different models exist for applying the Support Vector method to regression problems. The aim of a Support Vector Machine is to devise a computationally efficient way of learning good separating hyperplanes in a high dimensional feature space. [6]

C. K-Nearest Neighbor

K-nearest-neighbor (kNN) classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K-nearest-neighbor classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine. In an unpublished US Air Force School of Aviation Medicine report in 1951, Fix and Hodges introduced a non-parametric method for pattern classification that has since become known the k-nearest neighbor rule (Fix & Hodges, 1951). Later in 1967, some of the formal properties of the k-nearest-neighbor rule were worked out; for instance it was shown that for k=1 and $n \to \infty$ the k-nearest-neighbor classification error is bounded above by twice the Bayes error rate (Cover & Hart, 1967). Once such formal properties of k-nearest-neighbor classification were established, a long line of investigation ensued including new rejection approaches (Hellman, 1970), refinements with respect to Bayes error rate (Fukunaga & Hostetler, 1975), distance weighted approaches (Dudani, 1976; Bailey & Jain, 1978), soft computing (Bermejo & Cabestany, 2000) methods and fuzzy methods (Jozwik, 1983; Keller et al, 1985). [7]

V. ALGORITHM COMPARISON

In this section the three mentioned algorithms will be compared and descibed which parameters were used and how they affect the final score.

First of all, we are going to observe the data set and its size. As mentioned, data set has 8124 instances, the results we will present worked for traing set of 80% and test set of 20%. The size of test set between 20% and 60% works almost the same and does not affect the final score, where test data set size lower than 20% or higher than 60% gives lower performances.

TABLE I. SIMULATION RESULT OF ALGORITHM SVM

Num	**Support Vector Machines**			
	kernel	***C***	***gamma***	***F1 score***
1	linear	0.1	0.1	0,948667966212
2	linear	1	0.1	0,954721862872
3	linear	10	0.1	0,977099236641
4	rbf	0.1	0.1	0,997402597403
5	**rbf**	**1**	**0.1**	**1**
6	rbf	10	0.1	1

As you can see from the TABLE I. ‘rbf’ kernel works better than the linear for this data set. With the raise of the C parameter , which represents penalty parameter C of the error term, the score is also increasing. The C value for 1 or higher gives the same, the best score. Gamma is Kernel coefficient for ‘rbf’ kernel and we discovered that Gamma value of 0.1 works the best for our data set. The fifth row in TABLE I. presents the parameters we used in our project which gave the best results.

TABLE II. SIMULATION RESULT OF ALGORITHM KNN

Num	**K-Nearest Neighbors**			
	K	***p***	***metric***	***F1 score***
1	**5**	**2**	**minkowski**	**1**
2	7	2	minkowski	0,999352750809
3	10	2	minkowski	0,9980532122
4	3	1	minkowski	1

For KNN k value 5 and Euclidean distance gave us the best score, eventhough there is no big difference between Manhattan and Eucliedan distance. The performace is lower when the number of neighbours is bigger. For 5 and 6 values the F1 score is 1. The first row in TABLE II. presents the parameters we used in our project.

Since Naïve Bayes does not use any parameters, the result it gave is 0,929541047188.

VI. CONCLUSION

It can be seen from the results mentioned before that the SVM and KNN classification techniques perform the best among the three techniques used for classification. Performance of all techniques is lower when data set size is small and the performance improves with increase in size of training set. So it is very clear that size of training set, as well as the parameters and selection of classification technique depending on the data to be analysed, is very importatn for mining of patterns efficiently.

REFERENCES

[1] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.

[2] R. Duda, P. Hart, D. Stork, Pattern Classification, JWiley-Interscience, 2001.

[3] https://archive.ics.uci.edu/ml/datasets/mushroom

[4] A Tour of Machine Learning Algorithms by Jason Browniee

[5] https://medium.com/@gp_pulipaka/applying-gaussian-na%C3%AFve-bayes-classifier-in-python-part-one-9f82aa8d9ec4

[6] Hofmann, Martin. "Support Vector Machines -- Kernels and the Kernel Trick"

[7] Pal, S.K., Mitra, P. Pattern Recognition Algorithms for Data Mining: Scalability, Knowledge Discovery and Soft Granular Computing. Boca Raton (FL), Chapman & Hall, 2004.