Pune Institute of Computer Technology
Department of Computer Engineering
Academic Year 2021-22

DATA MINING & WAREHOUSING
MINI PROJECT REPORT ON

Predicting the age of abalone from physical measurements

Performed by
41415 Sanket Bhatlawande
41414 Samarth Bhadane
41402 Abhishek Sawalkar

Under the guidance of Prof.
K.C. Waghmare

Abstract:
In this mini project we intend to predict the age of abalone using the dataset provided.
The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

The main objective is to determine the age of Abalone from the physical measurements. In this, we will be analysing the predictions using a confusion matrix and predicting using SVC, decision tree, KNN and Gaussian Naïve Bayes as examples.

Software and Hardware Requirements:

Software Requirements: Linux OS, Anaconda/Jupyter notebook or Google Colab, python3 3.2

Hardware Requirements: Laptop with 8GB RAM, Intel processor

Introduction:

Prediction of the age of Abalone **Outcomes:**

1. We split the data using KFold technique.
2. We try building a model using SVC, Decision Tree, KNN and Gaussian NB.
3. We analyze the results of individual models using confusion matrices.

Objectives:

1. Learn how to apply preprocessing steps on a labelled dataset.
2. Learn to build various data prediction models.
3. Learn to split dataset into train and test set using KFold technique.
4. Learn analyzing results with confusion matrix.

Scope:

We can tune the hyperparameters of the models to get a better result using models like KNN, which with the default settings give less accurate results than SVC. This helps us understand how the different models act on the same data, giving us an idea where it is better to use which.

Dataset details:

The UCI Machine Learning Repository provides one dataset abalone.data, it contains 4177 observations, 8 descriptives features and 1 target feature.

The target feature is the rings of abalone. It is an integer to describe the age of abalone, number of rings add1.5 gives the age in years of them.

Descriptive Features:

The variable description is:

Name / Data Type / Measurement Unit / Description

Sex / nominal / -- / M, F, and I (infant)

Length / continuous / mm / Longest shell measurement

Diameter / continuous / mm / perpendicular to length

Height / continuous / mm / with meat in shell

Whole weight / continuous / grams / whole abalone

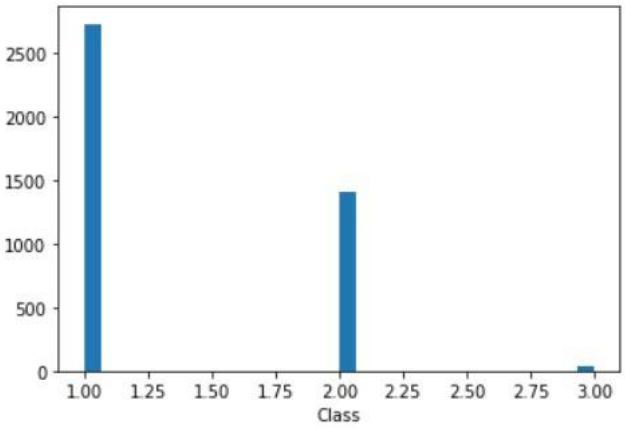
Shucked weight / continuous / grams / weight of meat

Viscera weight / continuous / grams / gut weight (after bleeding)

Shell weight / continuous / grams / after being dried

Rings / integer / -- / +1.5 gives the age in years\

A sample histogram depicting the data and the classes:

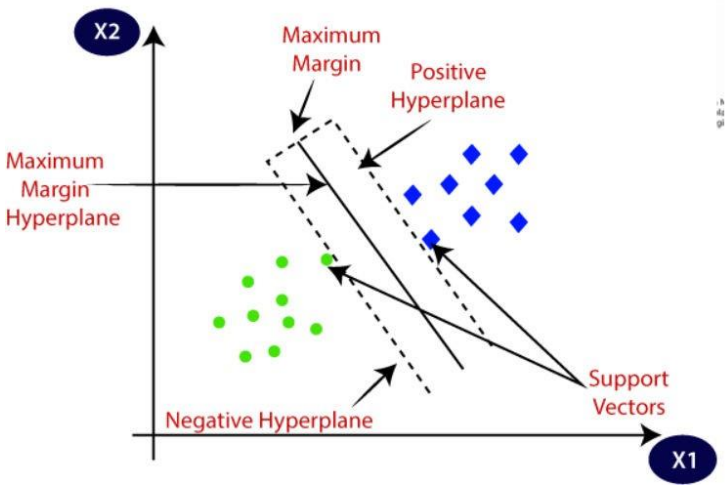


Theory:

Models analyzed:

1. SVC – support vector classifier:
   For understanding SVC, we need to also understand SVM which is Support Vector Machine.
   Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

   The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.



   The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations. Let's get started.

2. Gaussian Naïve Bayes:
   Naive Bayes is a classification algorithm for binary (two-class) and multiclass classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on.
Naive Bayes can be extended to real-valued attributes, most commonly by assuming a Gaussian distribution.

This extension of naive Bayes is called Gaussian Naive Bayes. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to work with because you only need to estimate the mean and the standard deviation from your training data.

Learn a Gaussian Naive Bayes Model from Data:
This is as simple as calculating the mean and standard deviation values of each input variable (x) for each class value.

mean(x) = 1/n * sum(x)

Where n is the number of instances and x are the values for an input variable in your training data.

We can calculate the standard deviation using the following equation:

standard deviation(x) = sqrt(1/n * sum(xi-mean(x)^2))

This is the square root of the average squared difference of each value of x from the mean value of x, where n is the number of instances, sqrt() is the square root function, sum() is the sum function, xi is a specific value of the x variable for the i'th instance and mean(x) is described above, and ^2 is the square.

3. KNN – K nearest neighbors:

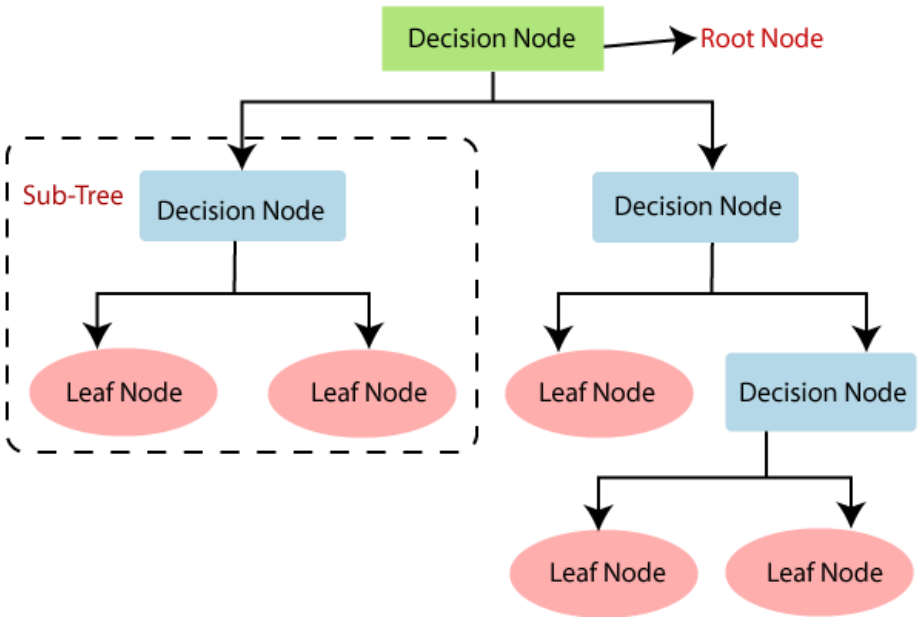
K Nearest Neighbor algorithm falls under the Supervised Learning category and is used for classification (most commonly) and regression. It is a versatile algorithm also used for imputing missing values and resampling datasets. As the name (K Nearest Neighbor) suggests it considers K Nearest Neighbors (Data points) to predict the class or continuous value for the new Datapoint.

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

The KNN Algorithm:
Load the data
Initialize K to your chosen number of neighbors
3. For each example in the data
3.1 Calculate the distance between the query example and the current example from the data.
3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

4. Decision tree:
Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have

multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
The decisions or the test are performed on the basis of features of the given dataset.
It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
Below diagram explains the general structure of a decision tree:



Steps for Decision Tree:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
Step-3: Divide the S into subsets that contains possible values for the best attributes.
Step-4: Generate the decision tree node, which contains the best attribute.
Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

5. Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

Basic terminologies:
true positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease. true negatives (TN): We predicted no, and they don't have the disease. false positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
false negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Example:

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Implementation details:

1. Load the data.
2. Visualize the data using histograms.
3. Implement the above models and check the accuracies and analyze the results using confusion matrix to look at the true positives and true negatives.
   Results obtained in terms of score: a.
   SVC – 76.6%
   b. KNN – 74.85%
   c. Gaussian NB – 63.3%
   d. Decision Tree – 69.8%

Model chosen – The SVC and KNN give very close accuracy and we hope to tune the hyperparameters for both to gain a final understanding of the models. For now, we go with Support Vector Classifiers.

Conclusion:

We have successfully created a prediction model for the Age of Abalone using given dataset and also analyzed various models on the same.