Abalone Age Group Classification

A Report Written in Partial Fulfillment
of the requirements of Course Project of EE 769

Submitted On:
5th June 2020

Submitted by:
Krishanu Sarkar
193170022

Under the guidance of:
Prof. Amit Sethi,
Department of Electrical Engineering ,
Indian Institute of Technology Bombay,
Mumbai - 400076

Introduction

Machine learning (ML) algorithms are used to recognize patterns and make decisions based on empirical data. The problem of classification of a data set, that is, assigning a class label to each sample of the data set can be complex, especially if the data set large or if the data set has a large feature dimension. If the feature vector representing a sample of data is of n dimensions, the problem of classification boils down to carving out regions in n-dimensional feature space with the understanding that any point within a space is to be assigned a certain class label. Abalones, also called ear-shells or sea ears, are sea snails (marine gastropod mollusks) found world-wide. The age of an abalone can be determined by counting the number of layers in its shell. However, age determination is a cumbersome process: It involves cutting a sample of the shell, staining it, and counting the number of rings through a microscope. A data set provided by the University of California Irvine Machine Learning Repository consists of physical characteristics of abalones and their ages. This study is a classification problem that aims to predict the age range of abalones using their physical characteristics. The data set is pre-processed to transform the problem of predicting the age to a classification problem.

Data-set:

History of the data-set:

The dataset comes from a 1994 study "The Population Biology of Abalone (Haliotis species) in Tasmania.
I. Blacklip Abalone (H. Rubra) from the North Coast and Islands of Bass Strait". This original data set contained missing values. But the data set provided in UCI machine learning repository has no missing value.

Brief Description Of the Data-set:

The University of California, Irvine Machine Learning Repository provides a data set consisting of 4177 samples of physical characteristics of abalones and their age. Abalones are sea-snails that are fished for their shells and meat. Scientific studies on abalones require knowing the age of an abalone, but the process of determining age is complicated. It involves measuring the number of layers of shell ("rings") that make up the abalone's shell. This is done by taking a sample of shell, staining it and counting the number of rings under the microscope. To circumvent the cumbersome process, this data set provided has been used to build learning algorithms to predict age using easily and quickly measurable physical characteristics. This project uses this data set recast as a classification problem, rather than a prediction problem.

The data set consists of 8 features and the number of rings (which is directly related to the age). The 8 features are sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight.

The number of rings varies from 1 to 29. This could be looked at as a class label that can take 29 possible values. To reduce the time taken by code to run experiments, the number of classes is reduced from 29 to 5. Each class forms an age range.

Problem Statement:

Description of the problem:

We have considered a data set which consists of physical measurements of abalones. Abalone is the name for a group of different sized sea snails of Haliotis family.

Contents of the problem:

The abalone data set contains 4177 rows and 9 columns. These columns include 7 continuous attributes (Length, Diameter, Height, Whole weight, Shucked weight, Viscera weight, Shell weight), a categorical attribute (Sex) which is divided into three categories and one integer response variable (Number of rings). So attribute(Sex) has three categories Male, Female, Infant ,by one hot encoding we replace the categories by three columns(Male, Female and Infant). If the abalone is male then we introduce 1 in the Male column and 0 in the columns of Female and Infant. We continue this approach similarly while the abalone is female or infant.

In given data set rings have value from 1 to 29 so divide it into 5 class, we classify it as :

Rings Range	Class
1-6	1
7-12	2
13-18	3
19-24	4
24-29	5

There are no missing value in this data set.

Type of the problem:

Original problem states that we have to predict the age of abalones from given physical measurements. The age is determined by cutting the shell through the cone, counting the number of rings through a microscope and also with other measurements which can be obtained easily to predict the age.age is calculated as Rings + 1.5 The problem provided in UCI machine learning repository is to fit regression models.

Original problem is regression problem But we convert this problem into a Classification Problem by dividing classes into 5 groups i.e. if abalone is of age between 1-6 years belongs to group 1, age between 7-12 years belongs to group 2 etc. The classification depends upon the data such that there are equal number of data points in each group.
Age of the Abalone is calculated as Rings+1.5

Overall approach to the problem:

Before applying any technique of this classification problem, we knock out 10 percent of the data and then we will use imputation techniques for these missing data values.
Applied Techniques:

Before applying various supervised learning techniques for classification, we will use imputation techniques to fill up the missing data values.

We will use the following imputation techniques to make up the missing

values:

Sr No.	Name of the Technique
1	Mean Value Imputation
2	Median Value Imputation
3	kNN Imputation

We will use the following techniques to solve the problem:

Sr No.	Name of the Technique
1	Decision Tree Classifier
2	Adaboost Classifier
3	KNN Classifier
4	Naive Bayes Classifier
5	SVM

Now we compare the outputs of the given algorithms and will find out the best among them.

Imputation Techniques for missing data

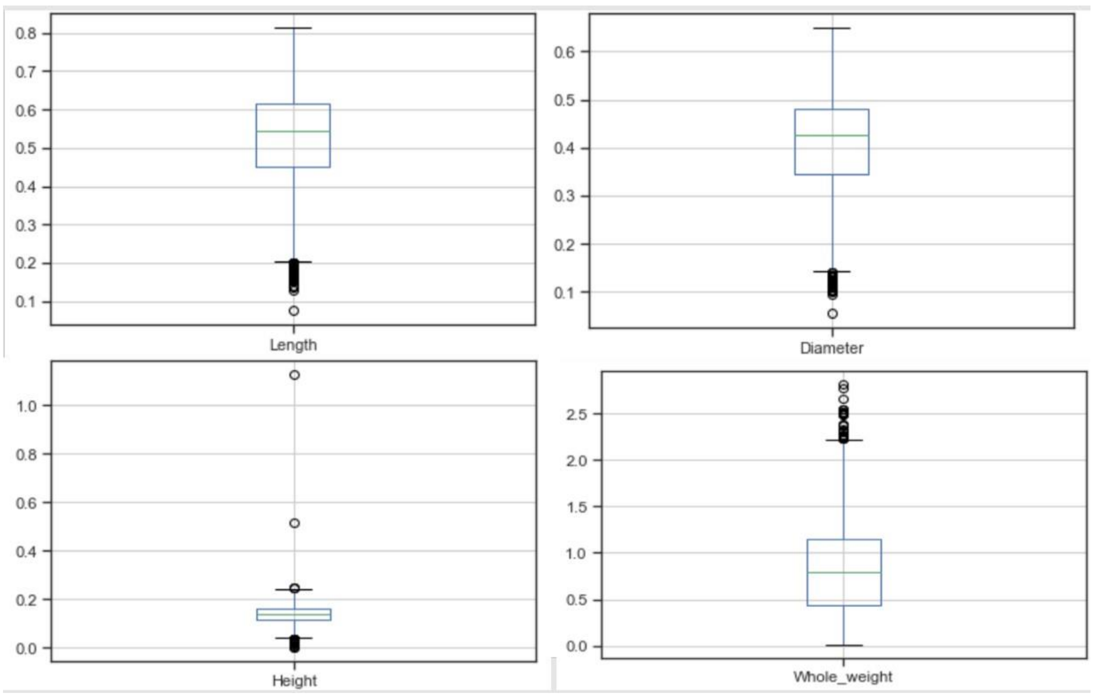
Imputation Using Mean Values:

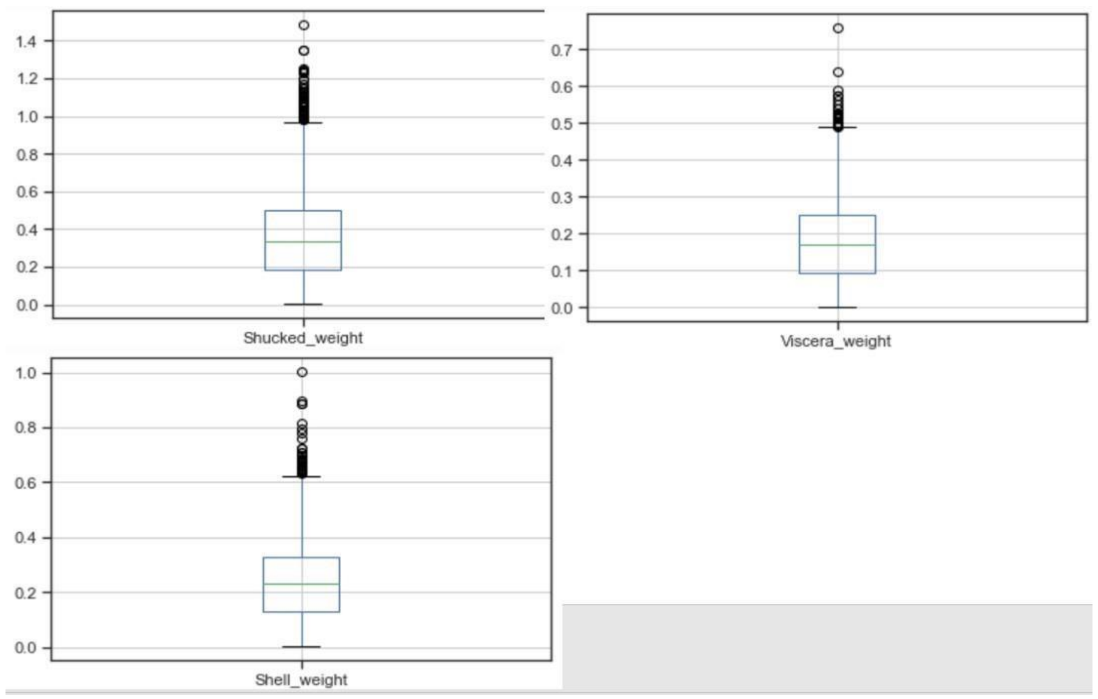
In this imputation technique, we will calculate the mean of the non-missing values in a column and then replacing the missing values within each column separately and independently from the others. It can only be used with numeric data. However, mean imputation attenuates any correlations involving the variable(s) that are imputed. This is because, in cases with imputation, there is guaranteed to be no relationship between the imputed variable and any other measured variables. Thus, mean imputation has some attractive properties for univariate analysis but becomes problematic for multivariate analysis.

Imputation Using Median Values:

Visualizing the columns of the data i.e. knockout data, we make boxplot for each column. In this case, we see that there are large outlier values so we use median of each column to fill the data in that column.

Boxplot for column values:





Imputation using KNN:

For large number of missing values, using mean or median can result in loss of variation in data. So we try KNN method for imputation.The assumption behind using KNN for missing values is that a point value can be approximated by the values of the points that are closest to it, based on other variables. KNN algorithm is useful for matching a point with its closest k neighbors in a multi-dimensional space. It can be used for data that are continuous, discrete, ordinal and categorical which makes it particularly useful for dealing with all kind of missing data.
The KNNImputer class provides imputation for completing missing values using the k-Nearest Neighbors approach. Each sample's missing values are imputed using values from n neighbors nearest neighbors found in the training set. Note that if a sample has more than one feature missing, then the sample can potentially have multiple sets of n neighbors donors depending on the particular feature being imputed. Each missing feature is then imputed as the average, either weighted or unweighted, of these neighbors. Where the number of donor neighbors is less than n neighbors, the training set average for that feature is used for imputation. The total number of samples in the training set is, of course, always greater than or equal to the number of nearest neighbors available for imputation, depending on both the overall sample size as well as the number of samples excluded from nearest neighbor calculation because of too many missing feature.

Various Classifications Techniques Used For Abalone Age Group Prediction

Decision Tree Classifier:

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems too. We use Decision Tree algorithm to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data(training data).

The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

In this algorithm, first we used to place the best attribute of the dataset at the root of the tree. Then we split the training set into subsets in such a way that each subset contains data with the same value for an attribute. Then we repeat this method on each subset until we find leaf nodes in all the branches of the tree.

In decision trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

We continue comparing our record's attribute values with other internal nodes of the tree until we reach a leaf node with predicted class value. As we know how the modeled decision tree can be used to predict the target class or the value. Now let's understanding how we can create the decision tree model.

When used on dataset imputed by mean values:

Accuracy = 0.7535885167464115

When used on dataset imputed by median values:

Accuracy = 0.7535885167464115

When used on dataset imputed by kNN regression:

Accuracy = 0.7727272727272727

Adaboost Classifier:

Adaboost, like Random Forest Classifier is another ensemble classifier. Ensemble classifier are made up of multiple classifier algorithms and whose output is combined result of output of those classifier algorithm.

Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.

AdaBoost, short for “Adaptive Boosting”, is the first practical boosting algorithm proposed by Freund and Schapire in 1996. It focuses on classification problems and aims to convert a set of weak classifiers into a strong one. The final equation for classification can be represented as

$$F(x) = \text{sign}\left(\sum_{m=1}^{M} \theta_m f_m(x)\right)$$

Where, f_m stands for the m-th weak classifier and and θ_m is the corresponding weight. It is exactly the weighted combination of M weak classifiers.

Generally, AdaBoost is used with short decision trees. Further, the first tree is created, the performance of the tree on each training instance is used. Also, we use it to weight how much attention the next tree. Thus, it is created should pay attention to each training instance. Hence, training data that is hard to predict is given more weight. Although, whereas easy to predict instances are given less weight.

kNN Classifier:

kNN or k-nearest neighbours is the simplest classification algorithm. This classification algorithm does not depend on the structure of the data. Whenever a new example is encountered, its k nearest neighbours from the training data are examined. Distance between two examples can be the euclidean distance between their feature vectors. The majority class among the k nearest neighbours is taken to be the class for the encountered example.

kNN classifier algorithm finds relations between predictors and outcomes and the relationships are summarised in a model.Then we test the model on a test sample whose class labels are known but not used for training the model. Now we use the model for classification on new data whose class labels are unknown.

We consider the similarity of two points to be the distance between them in this space under some appropriate metric. The way in which the algorithm decides which of the points from the training set are similar enough to be considered when choosing the class to predict for a new observation is to pick the k closest data points to the new observation, and to take the most common class among these. This is why it is called the k Nearest Neighbours algorithm.

In this algorithm, we first specify a positive integer k along with a new sample. Then we select the k entries in our database which are closest to the new sample. Thus we find the most common classification of these entries.

Naive Bayes Classifier:

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. It assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems.

Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data Here we have used the Gaussian Naive Bayes classifier.

When used on dataset imputed by mean values:

Accuracy = 0.4569357990460622

When used on dataset imputed by median values:

Accuracy = 0.4569377990430622

When used on dataset imputed by kNN regression:

Accuracy = 0.43779904306220097

Support Vector Machine:

SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces.

The SVM algorithm is implemented in practice using a kernel. A kernel transforms an input data space into the required form. SVM uses a technique called the kernel trick. Here, the kernel takes a low-dimensional input space and transforms it into a higher dimensional space. In other words, you can say that it converts non-separable problem to separable problems by adding more dimension to it. It is most useful in nonlinear separation problem. Kernel trick helps you to build a more accurate classifier.

Linear Kernel

A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.

Polynomial Kernel

A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space.

Radial Basis Function Kernel

The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification. RBF can map an input space in infinite dimensional space.

Hyperparameter Tuning Used in The Above Classification Techniques:

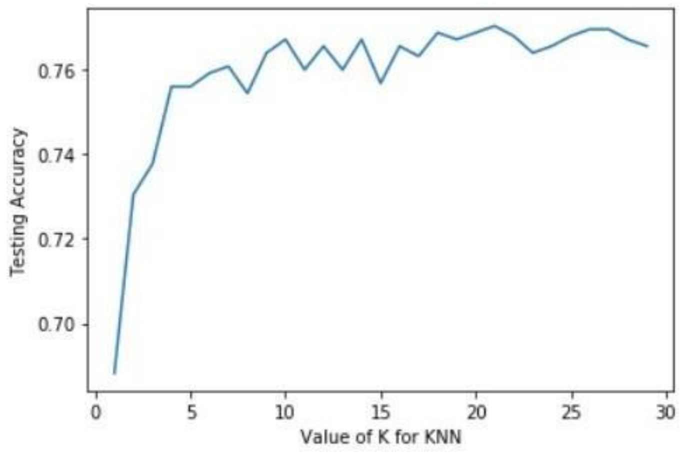
Hyperparameter Tuning Used For kNN regression:

Here, the hyperparameters are

- The value of K
- The value of p used in L^p norm for calculating distances between points When used on

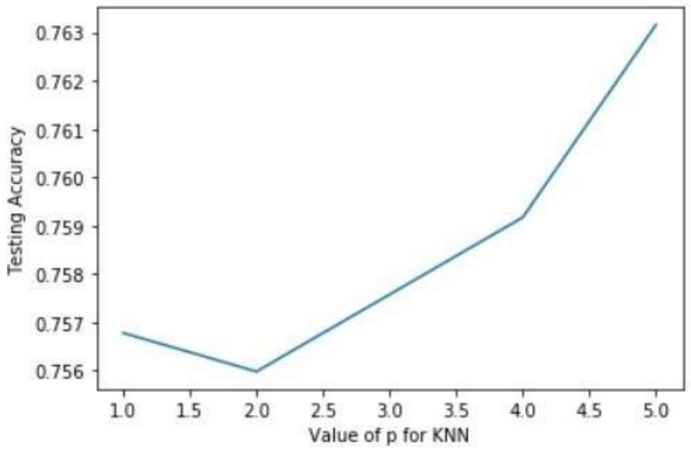
dataset imputed by mean and median are almost equal values:

Best value of K =21
Accuracy before tuning=
Accuracy after tuning = 0.77033492

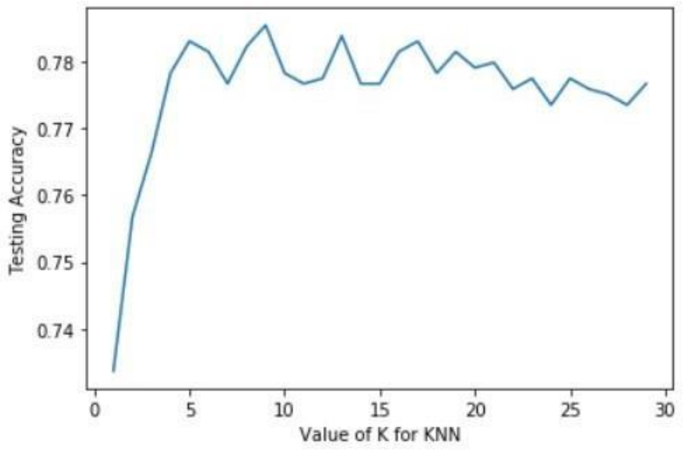


Accuracy for different distance
best distance is 5 Accuracy before
tuning =

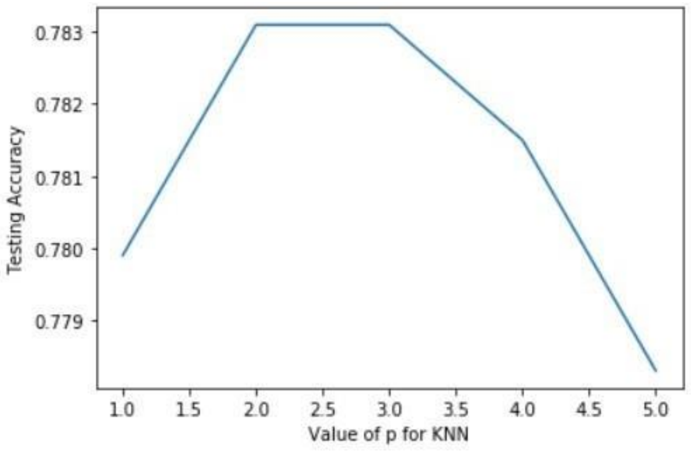
Accuracy after tuning = 0.7631578947368421



When used on dataset imputed by kNN regression: best value of K =9 Accuracy before tuning=
Accuracy after tuning = 0.7854864



Accuracy for different distance best distance is p = 2 and p=3
Accuracy before tuning =
Accuracy after tuning = 0.7631578947368421

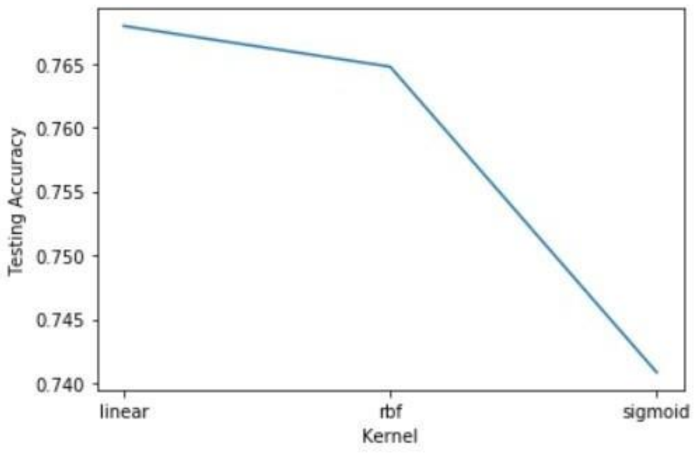


Hyperparameter Tuning Used For Support Vector Machine:

Here, the hyperparameters are

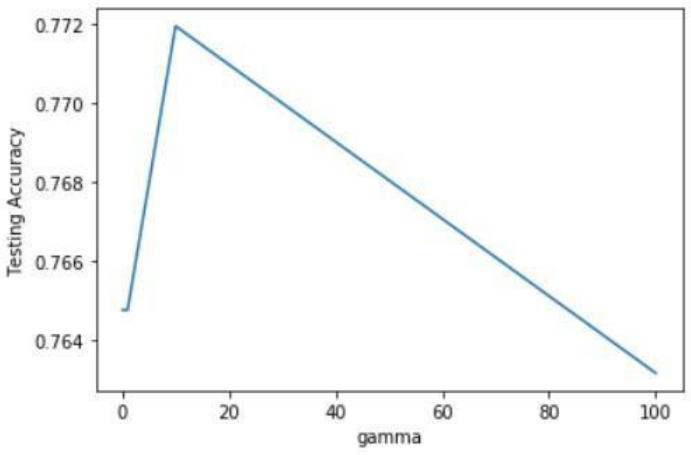
- Different kernels used in SVC classifier
- The values of parameters γ and C used in RBF kernel

For mean Imputed data For different
kernels best Accuracy is for kernel = linear is
0.7679

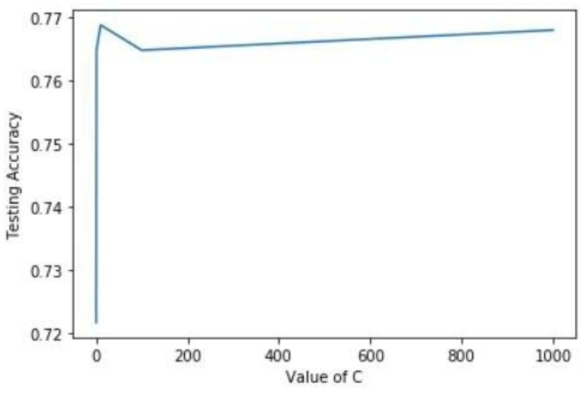


For different gamma

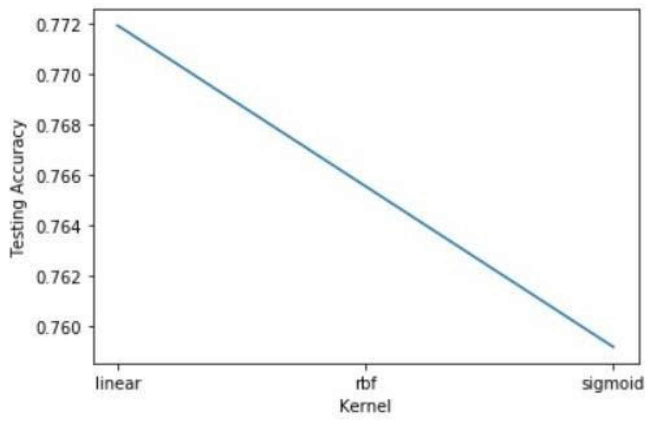
best Accuracy is for gamma = 10 is 0.7719 For different cs



best Accuracy is for cs = 10 is 0.7687

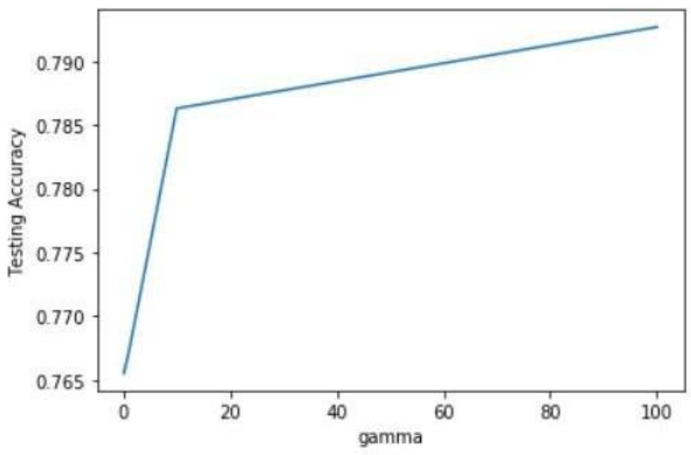


For Median Imputed data
Accuracies for mean median are almost equal
For KNN Imputed data For different kernels
best Accuracy is for kernel = linear is 0.7719

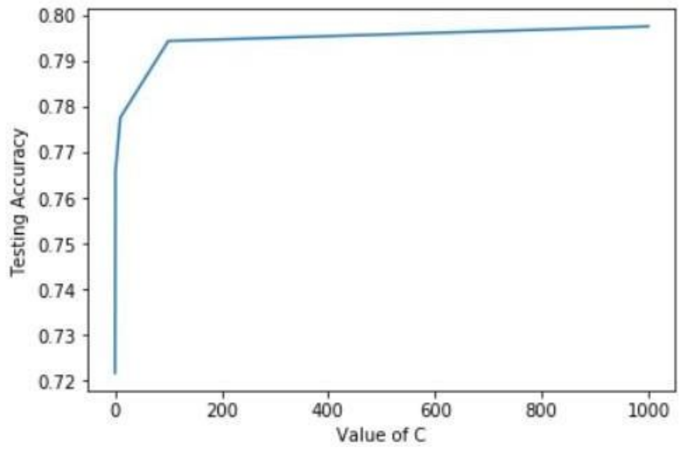


For different gamma

best Accuracy is for gamma = 100 is 0.7926 For different cs



best Accuracy is for cs = 1000 is 0.7974



Hyperparameter Tuning Used For Adaboost Classifier:

Here, the hyperparameters are

- algorithm for fitting
- best estimator of the classifier
- number of estimators
- learning rate

When used on dataset imputed by mean values:

Accuracy before tuning = 0.6499202551834131

Accuracy after tuning = 0.7472089314194578

When used on dataset imputed by median values:

Accuracy before tuning = 0.6499202551834131

Accuracy after tuning = 0.7472089314194578

When used on dataset imputed by kNN regression:

Accuracy before tuning =

0.5542264752791068 Accuracy after tuning =

0.766347687400319

Comparison Of Various Techniques Of Imputation:

After looking at the accuracy obtained for each classifier we observe that the imputation techniques using mean and median values in filling missing values in the dataset gives almost same result. But we observe that the imputed values are far enough from actual values. But if we estimate the missing values using kNN regression then the estimated values are close enough to actual. Hence we can conclude kNN regression is the best technique for imputation in case of our dataset.

Comparison Of Various Techniques Of Classification:

After looking at the accuracy and Confusion matrix obtained for each classifier we observe that the KNN and SVM Classifier work best for the Abalone age group prediction purpose after we have done hyperparameter tuning.

Conclusion

Artificial Intelligence (AI) is everywhere. Possibility is that you are using it in one way or the other and you don't even know about it. One of the popular applications of AI is Machine Learning (ML), in which computers, software, and devices perform via cognition (very similar to human brain). Herein, we share few examples of machine learning that we use everyday and perhaps have no idea that they are driven by ML.

We will use the following imputation techniques to make up the missing

values:

Sr No.	Name of the Technique
1	Mean Value Imputation
2	Median Value Imputation
3	kNN Imputation

- There are various to impute the data
- Mean and Median are the guess of the missing values by looking at the data
- We need strong techniques for imputation which are not based on the guess
- KNN is the good techniques to impute the data
- If we look at the imputed values we will see that K-nn gives closet prediction to original value • If

  we rank the Accuracy of techniques we will get : *mean < median < KNN*

We will use the following techniques to solve the problem:

Sr No.	Name of the Technique
1	Decision Tree Classifier
2	Adaboost Classifier
3	KNN Classifier
4	Naive Bayes Classifier

- In this project we applied various ML algorithms for our classification task
- We also use hyper-parametric tuning to improve the Accuracy
- We get a good result by doing hyper-parametric tuning accuracy is increased by 10 - 20 percent