

A PROJECT REPORT ON

**Fruit\_dataset\_with\_colours Using Machine Learning**

by

# Miss.Priya R.Umate

**B.E(CE)**

Batch: 2021-5127

Center: chinchwad

## Under the Guidance of,

Mr. Mathivanan Balakrishnan

Technical Trainer

### EduBridge

(School of Coding)

#### Introduction

#### 

Regression analysis is a set of statistical methods used for the estimation of relationships between a dependent variable and one or more independent variablesIndependent VariableAn independent variable is an input, assumption, or driver that is changed in order to assess its impact on a dependent variable (the outcome .

Regression analysis is primarily used for two conceptually distinct purposes. First, regression analysis is widely used for [prediction](https://en.wikipedia.org/wiki/Prediction) and [forecasting](https://en.wikipedia.org/wiki/Forecasting), where its use has substantial overlap with the field of [machine learning](https://en.wikipedia.org/wiki/Machine_learning). Second, in some situations regression analysis can be used to infer [causal relationships](https://en.wikipedia.org/wiki/Causality) between the independent and dependent variables. Importantly, regressions by themselves only reveal relationships between a dependent variable and a collection of independent variables in a fixed dataset. To use regressions for prediction or to infer causal relationships, respectively, a researcher must carefully justify why existing relationships have predictive power for a new context or why a relationship between two variables has a causal interpretation. The latter is especially important when researchers hope to estimate causal relationships using [observational data](https://en.wikipedia.org/wiki/Observational_study).

Machine learning:--

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

**K-Nearest Neighbors algorithm**

the ***k*-nearest neighbors algorithm** (***k*-NN**) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) [classification](https://en.wikipedia.org/wiki/Classification) method first developed by [Evelyn Fix](https://en.wikipedia.org/wiki/Evelyn_Fix) and [Joseph Hodges](https://en.wikipedia.org/wiki/Joseph_Lawson_Hodges_Jr.) in 1951,[[1]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-1) and later expanded by [Thomas Cover](https://en.wikipedia.org/wiki/Thomas_M._Cover).[[2]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-2) It is used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). In both cases, the input consists of the *k* closest training examples in [data set](https://en.wikipedia.org/wiki/Data_set). The output depends on whether *k*-NN is used for classification or regression:

* In *k-NN classification*, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In *k-NN regression*, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors.

*k*-NN is a type of [classification](https://en.wikipedia.org/wiki/Classification) where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then [normalizing](https://en.wikipedia.org/wiki/Normalization_(statistics)) the training data can improve its accuracy dramatically.[[3]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-:0-3)[[4]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-4)

Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/*d*, where *d* is the distance to the neighbor.[[5]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-5)

The neighbors are taken from a set of objects for which the class (for *k*-NN classification) or the object property value (for *k*-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

###### **Data Processing**

Itis the process of performing data operations to collect, convert and indentify data to generate useful informations. In this lab session we are going to handle noisy, inconsistent, intentional data using python library which can handle various type of encoding such as comma-separated values (CSV), eXtensible Markup Language (XML), Hyper Text Markup Langauge (HTML), Structured Query Language (SQL), JavaScript Object Notation (JSON) etc. For this encoding process different types of modules should be imported. Python's has different data pre-processing library where Pandas is one of them langauge package used for data processing.

**Python Data Preprocessing methods.**

* Importing the libraries
* Importing the dataset
* Handling the Missing data
* Split the dataset into training and testing datasets.
* Feature Scaling

**Important Libraries need to be imported for ML projects:**

* **Pandas** : Pandas is the most popular and favourite data science library written in the Python Programming Language for data manipulation and analysis also pandas provides fast analysis as well as data cleaning and preparation. The best is that Pandas can work with variety of data such as: Excel Sheet, csv file, SQL file or even a webpage. Some of the features using Python Pandas library are listed below: 1. Pandas Data Frame makes manipulating data easy, we are able to select, replace columns and rows and even reshape our data. 2. Pandas allows to perform conditional selecting conditional selection using bracket notation []. 3. Pandas allow index of a Data Frame 4. Pandas allow setting the index of a Data Frame. 5. Pandas will automatic fill in those missing points with a NaN or Null value, also we can replace our missing values using .fillna() method. 6. Pandas has a .groupby () method which is used to group together rows based off a column so that we can perform aggregate functions (sum, mean. median, standard deviation) 7. Pandas allows to get the number of times occurs in a DataFrame. 8. Pandas has .describe() method is used to get an overview of DataFrame. 9.Similarly , we can concatenate, merge and join multiple DataFrame,
* **Numpy** : Numpy is a package used for scientific calculating and perform various operations. Numpy Array is a multidimensional array whose indexed is similar to Sequences which is start with zero that is used to store values of same datatype. Numpy in the python uses less memory to store data as compared with python list. Numpy provides multiple functions they are nonzero and count\_non-zero for finding the element. Using Numpy Library we can speed up our workflow and interface with other package in python ecosystem.
* **Matplotlib** : Malplotlib which was introduced by John Hunter is a multiplatform data visualization library built on Numpy arrays and designed to work with broader SciPy stack. Similarly, matplotlib.pyplot is a collection of command style functions that make matplotlib work like MATLAB. Pyplot is mainly intented for interative plots and simple cases of programmatic plot generations.
* **ScikitLearn** : Scikit-learn (formerly scikits.learn and also known as sklearn) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language). It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy).
* **Seaborn**: One of the best but also more challenging ways to get your insights across is to visualize them: that way, you can more easily identify patterns, grasp difficult concepts or draw the attention to key elements. When you’re using Python for data science, you’ll most probably will have already used [Matplotlib](https://matplotlib.org/), a 2D plotting library that allows you to create publication-quality figures. Another complimentary package that is based on this data visualization library is [Seaborn](http://seaborn.pydata.org/), which provides a high-level interface to draw statistical graphics.

Import seaborn as sns

* **Count plot**

 A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable. The basic API and options are identical to those for barplot() , so you can compare counts across nested variables.

* **Histogram**

A histogram is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range. A histogram displays the shape and spread of continuous sample data.

* **Box plot**

In descriptive statistics, a box plot or boxplot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box.

* **Scatter plot**

A scatter plot (also called a scatterplot, scatter graph, scatter chart, scattergram, or scatter diagram)[[3]](https://en.wikipedia.org/wiki/Scatter_plot#cite_note-3) is a type of [plot](https://en.wikipedia.org/wiki/Plot_(graphics)) or [mathematical diagram](https://en.wikipedia.org/wiki/Mathematical_diagram) using [Cartesian coordinates](https://en.wikipedia.org/wiki/Cartesian_coordinate_system) to display values for typically two [variables](https://en.wikipedia.org/wiki/Variable_(mathematics)) for a set of data. If the points are coded (color/shape/size), one additional variable can be displayed. The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the [vertical axis](https://en.wikipedia.org/wiki/Vertical_axis).[[4]](https://en.wikipedia.org/wiki/Scatter_plot#cite_note-4)

**Process involved**:

* Importing the dataset
* To know whether any cell value is empty or not
* If empty, Filling missing value
* Preparing our data- defining attributes and labels
* The next step is to split this data into training and test sets
* Training the algorithm with one of the ML concept
* Import regression using Scikit-Learn library
* Making prediction based on attributes(inputs)
* Evaluating the model

**Languages Used:** Python

**Libraries Used:** Pandas, Numpy, Matplotlib, ScikitLearn, Seaborn.

**IDE** **Used**: Jupyter Notebook

**Coding and Explanation**

**KNN Algorithm**

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import train\_test\_split

X = fruit[['mass','width','height','color\_score']]

Y = fruit['fruit\_name']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)

X\_train.describe()

knn = KNeighborsClassifier()

knn.fit(X\_train, y\_train)

knn.score(X\_test, y\_test)

print("Accuracy for K=5 : ", knn.score(X\_test, y\_test))

knn = KNeighborsClassifier(n\_neighbors = 6)

knn.fit(X\_train, y\_train)

knn.score(X\_test, y\_test)

print("Accuracy for K=6 : ", knn.score(X\_test, y\_test))

knn = KNeighborsClassifier(n\_neighbors = 7)

knn.fit(X\_train, y\_train)

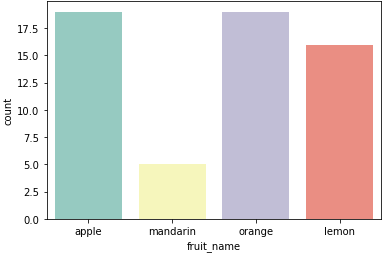
knn.score(X\_test, y\_test)

print("Accuracy for K=7 : ", knn.score(X\_test, y\_test))

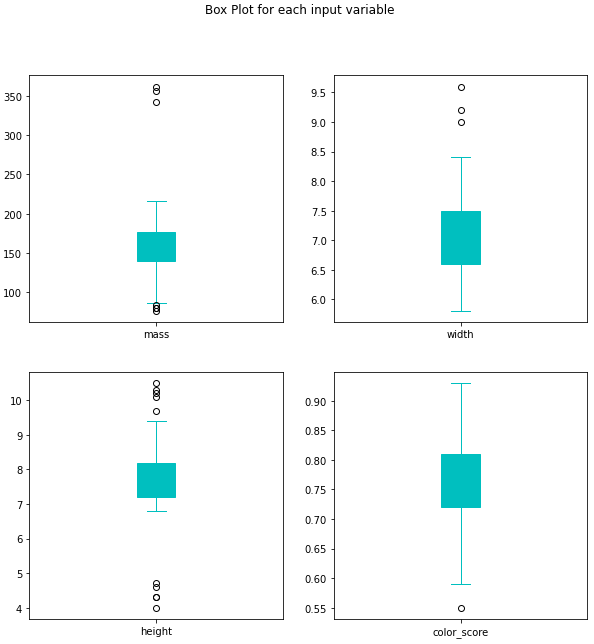
**Explaination**:--

* 1. In this first we importing the csv file with (.csv) extension.
  2. Then after importing csv file the training set test set splits.
  3. After that we have to use feature scalling
  4. After feature scalling we have to create linear model with traing sets.
  5. After that predicted the model
  6. Comparison: In this module, we introduce K-Nearest Neighbors (KNN), which is a non-parametric method
  7. Easy to fit. One needs to estimate a small number of coefficients. I Often easy to interpret.

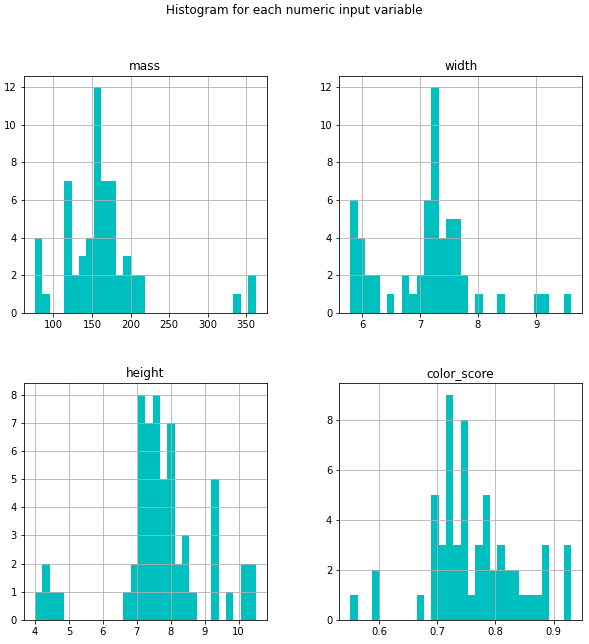
**Screen shots**

****

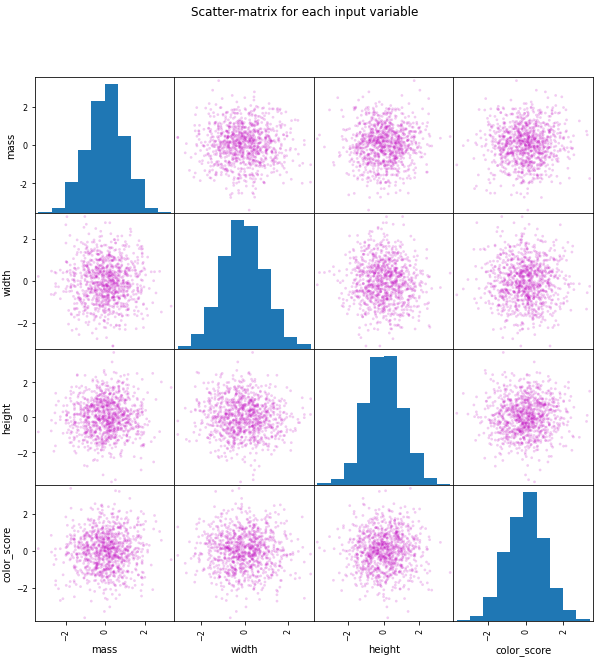
**Src: fruite dataset with color countplot**

****

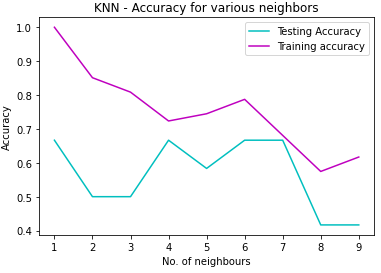
**Src: fruit dataset with color of Box Plot**

****

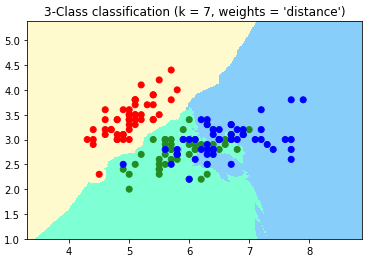
**Src: fruit dataset with color of histogram**

****

**Src: fruit dataset with color of Scatter plot**

****

**Src: fruit dataset with color of KNN accuracy**

****

**Src: fruit dataset with color of Classification chart**

# THANK YOU