Big Data Project Report

(Practical)

Surya Nikhil Mallampalli
AP20110010752

Gorantla Geetha Gayathri
AP20110010748

Tejaswini Pothuri
AP20110010728

Vasireddy Pooja Vishali
AP20110010770

Konathala Pavan Rahul
AP20110010746

Abstract

These models on a Cancer Dataset from kaggle, predicts whether the given characteristics belongs to Malignant or Benign Tumor for each Tuple.

Based on the Above type of classification we make based on the input data, the patient get treated for his respected Tumor Data, not only for the treatment Issue but also it matters a lot to classify his characteristics.

In this Project we have used we have used Naive Bayes Model, KNN Model, Logistic Regression Model, Random Forest Model.

As a conclusion we have got the best model as Random forest to classify our data with an Accuracy of 95.9%. so, we can use Random forest Classifier to classify the Tumor from the given characteristic Data of a Patient efficiently.

Introduction

In this Project we are going to predict whether the cancer is benign (noncancerous) or malignant (cancerous).**Benign tumors tend to grow slowly and do not spread. Malignant tumors can grow rapidly, invade and destroy nearby normal tissues, and spread throughout the body**. So, we are going to use Naive Bayes Classification Algorithm on the cancer dataset to Classify the Malignant and Benign cancer.

Description of dataset

Source-https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

n the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu

cd math-prog/cpo-dataset/machine-learn/WDBC/

Also can be found on UCI Machine Learning Repository:
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Attribute Information:

1) ID number

2) Diagnosis (M = malignant, B = benign)

3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)

f) compactness (perimeter^2 / area - 1.0)

g) concavity (severity of concave portions of the contour)

h) concave points (number of concave portions of the contour)

i) symmetry

j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three

largest values) of these features were computed for each image,

resulting in 30 features. For instance, field 3 is Mean Radius, field

13 is Radius SE, field 23 is Worst Radius.

All feature values are recoded with four significant digits.

Missing attribute values: none

Class distribution: 357 benign, 212 malignant

Procedures we followed

Preprocessing

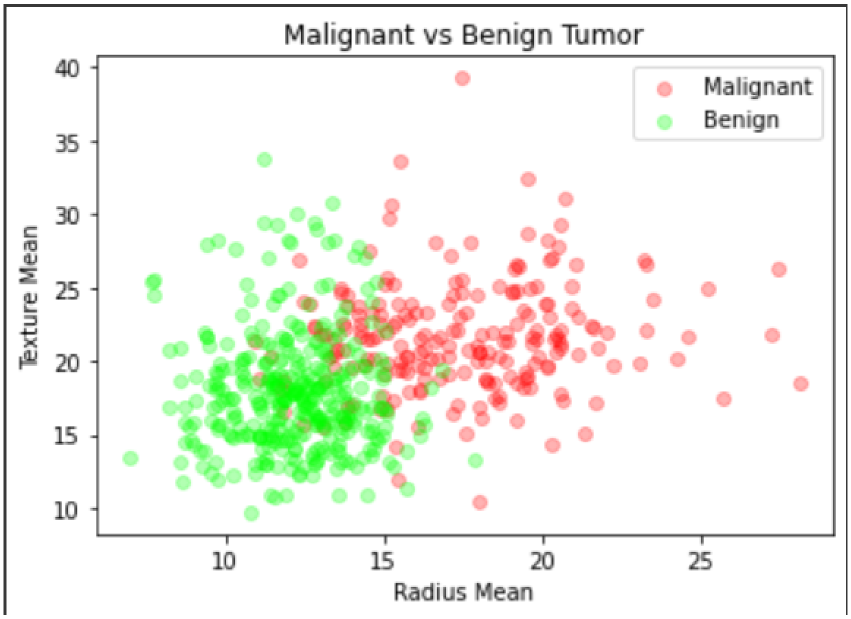
-The first phase was preprocessing the dataset. So, we removed all the unnecessary fields in the datasets like `"id"` and `"Unnamed: 32"` as these features are not needed while diagnosing whether the patient has cancer or not.

-After the preprocessing of the dataset, we only want to check the diagnosis which are Malignant and Benign Tumors.

-Then we make a data visualization of Malignant and Benign Tumors where the Texture Mean and Radius Mean are the labels of the scatterplot.

-We have used min-max normalization technique

Visualizing the Data



Here red color points in the plot represent Malignant Cancer, green color points represent Benign Tumor. This is plotted based on the Input Dataset we have taken.

Dividing Data

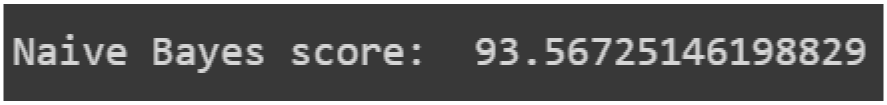
-Now we train the dataset, by changing the values of Malignant Cancer to 1 and Benign Cancer as 0 and apply min-max normalization using only 30% of the data for training it.

Related work:

Naive Bayes Model

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

The accuracy we got on the data set is: 93.5%

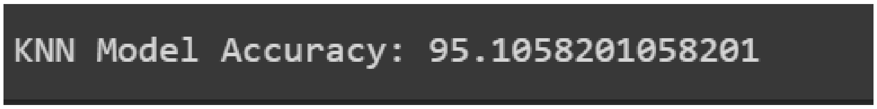


```
Naive Bayes score:  93.56725146198829
```

KNN Model

K-Nearest Neighbors Algorithm. The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

The accuracy we got on the data set is: 95.1%

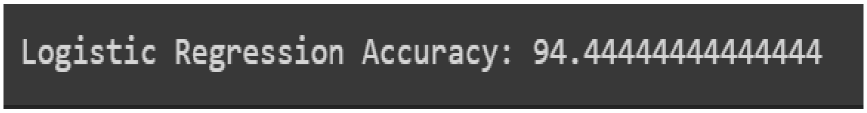


```
KNN Model Accuracy: 95.1058201058201
```

Logistic Regression Model

Logistic Regression is **a classification technique used in machine learning**. It uses a logistic function to model the dependent variable. The dependent variable is dichotomous in nature, i.e. there could only be two possible classes (eg.: either the cancer is malignant or not).

The accuracy we got on the data set is: 94.4%

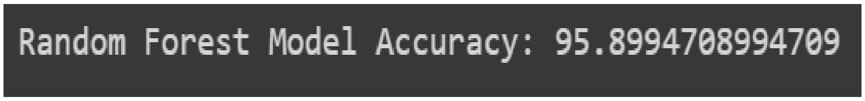


```
Logistic Regression Accuracy: 94.44444444444444
```

Random Forest Classifier Model

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the `max_samples` parameter if `bootstrap=True` (default), otherwise the whole dataset is used to build each tree.

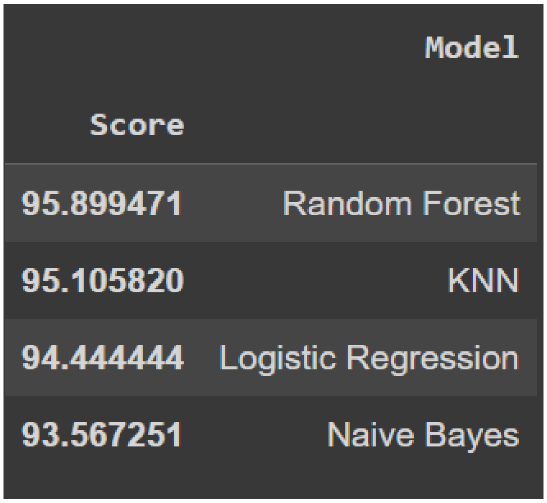
The accuracy we got on the data set is: 95.8%



```
Random Forest Model Accuracy: 95.8994708994709
```

Conclusion

After comparing the accuracy scores of different classification models like Logistic Regression, Random Forest Classifier, K neighbors Classifier with Naive Bayes Classifier. We concluded that Random Forest Classification has the highest accuracy followed by KNN ,Naive Bayes and Logistic Regression respectively.



	Model
Score	
95.899471	Random Forest
95.105820	KNN
94.444444	Logistic Regression
93.567251	Naive Bayes

Implementation:
https://colab.research.google.com/drive/1LrDB-39y__sXKqgDy0hVx1kDn82ZAEDK?usp=sharing

References

1. Implementation Code reference: https://colab.research.google.com/drive/1LrDB-39y__sXKqgDy0hVx1kDn82ZAEDK?usp=sharing
2. Understanding Data Preprocessing: https://towardsdatascience.com/data-preprocessing-e2b0bed4c7fb
3. Naive Bayes Model Doc Reference: https://scikit-learn.org/stable/modules/naive_bayes.html
4. KNN Model Doc Reference: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
5. Logistic Regression Doc Reference: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
6. Random Forest Doc Reference: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html