Analysis of different classification models

Code

Proposal

AUTHOR
Visual Voyagers

AFFILIATION

High level goal

To visualize different Machine Learning Classification Algorithms which can help identify whether a particular tumor is malignant or benign.

Dataset

Source: The data set we choose for this project is “Breast Cancer Diagnostic Data” we are getting the data set from UCI Machine Learning
Repository: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

The source of dataset https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Why this Dataset ?

The **Breast Cancer Wisconsin Diagnostic Dataset** is a widely used, high-quality dataset containing **569 samples** with **30 features** that describe tumor characteristics, allowing for effective **benign vs. malignant** binary classification. Its clinical relevance, well-labeled structure, and feature variety make it ideal for developing and benchmarking diagnostic models in machine learning.

This dataset is characterized by its **low dimensionality**, making it particularly well-suited for testing various classification algorithms and benchmarking their performance. It provides a reliable foundation for researchers and students to build predictive models, contributing to advancements in medical diagnostics and offering a valuable educational tool.

Basic EDA

```
library(ggplot2)
library(dplyr)

your_dataset <- read.csv("data/cancer_dataset.csv")

# View the first few rows of the dataset
head(your_dataset)
```

```
        id diagnosis radius_mean texture_mean perimeter_mean area_mean
1   842302         M       17.99        10.38         122.80    1001.0
2   842517         M       20.57        17.77         132.90    1326.0
3 84300903         M       19.69        21.25         130.00    1203.0
4 84348301         M       11.42        20.38          77.58     386.1
```

```
5 84358402         M       20.29        14.34         135.10    1297.0
6   843786         M       12.45        15.70          82.57     477.1
  smoothness_mean compactness_mean concavity_mean concave.points_mean
1         0.11840          0.27760         0.3001             0.14710
2         0.08474          0.07864         0.0869             0.07017
3         0.10960          0.15990         0.1974             0.12790
4         0.14250          0.28390         0.2414             0.10520
5         0.10030          0.13280         0.1980             0.10430
6         0.12780          0.17000         0.1578             0.08089
  symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
1        0.2419                0.07871    1.0950     0.9053        8.589
2        0.1812                0.05667    0.5435     0.7339        3.398
3        0.2069                0.05999    0.7456     0.7869        4.585
4        0.2597                0.09744    0.4956     1.1560        3.445
5        0.1809                0.05883    0.7572     0.7813        5.438
6        0.2087                0.07613    0.3345     0.8902        2.217
  area_se smoothness_se compactness_se concavity_se concave.points_se
1  153.40      0.006399        0.04904      0.05373           0.01587
2   74.08      0.005225        0.01308      0.01860           0.01340
3   94.03      0.006150        0.04006      0.03832           0.02058
4   27.23      0.009110        0.07458      0.05661           0.01867
5   94.44      0.011490        0.02461      0.05688           0.01885
6   27.19      0.007510        0.03345      0.03672           0.01137
  symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
1     0.03003             0.006193        25.38         17.33          184.60
2     0.01389             0.003532        24.99         23.41          158.80
3     0.02250             0.004571        23.57         25.53          152.50
4     0.05963             0.009208        14.91         26.50           98.87
5     0.01756             0.005115        22.54         16.67          152.20
6     0.02165             0.005082        15.47         23.75          103.40
  area_worst smoothness_worst compactness_worst concavity_worst
1     2019.0           0.1622            0.6656          0.7119
2     1956.0           0.1238            0.1866          0.2416
3     1709.0           0.1444            0.4245          0.4504
4      567.7           0.2098            0.8663          0.6869
5     1575.0           0.1374            0.2050          0.4000
6      741.6           0.1791            0.5249          0.5355
  concave.points_worst symmetry_worst fractal_dimension_worst  X
1               0.2654         0.4601                 0.11890 NA
2               0.1860         0.2750                 0.08902 NA
3               0.2430         0.3613                 0.08758 NA
4               0.2575         0.6638                 0.17300 NA
5               0.1625         0.2364                 0.07678 NA
6               0.1741         0.3985                 0.12440 NA
```

```r
# Get a summary of the dataset
summary(your_dataset)
```

```
       id             diagnosis          radius_mean      texture_mean   
 Min.   :     8670   Length:569         Min.   : 6.981   Min.   : 9.71  
 1st Qu.:   869218   Class :character   1st Qu.:11.700   1st Qu.:16.17  
 Median :   906024   Mode  :character   Median :13.370   Median :18.84  
 Mean   : 30371831                      Mean   :14.127   Mean   :19.29  
```

```
3rd Qu.:  8813129                    3rd Qu.:15.780   3rd Qu.:21.80
Max.   :911320502                    Max.   :28.110   Max.   :39.28
perimeter_mean     area_mean      smoothness_mean   compactness_mean
Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.   :0.01938
1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492
Median : 86.24   Median : 551.1   Median :0.09587   Median :0.09263
Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean   :0.10434
3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040
Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540
concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean
Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154
Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
  radius_se        texture_se      perimeter_se       area_se
Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530
Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337
3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190
Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200
smoothness_se      compactness_se      concavity_se     concave.points_se
Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.000000
1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638
Median :0.006380   Median :0.020450   Median :0.02589   Median :0.010930
Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   Mean   :0.011796
3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710
Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.052790
 symmetry_se       fractal_dimension_se  radius_worst   texture_worst
Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02
1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08
Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41
Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68
3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72
Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
perimeter_worst    area_worst     smoothness_worst  compactness_worst
Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729
1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720
Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190
Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427
3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910
Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800
concavity_worst  concave.points_worst symmetry_worst   fractal_dimension_worst
Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
Median :0.2267   Median :0.09993      Median :0.2822   Median :0.08004
Mean   :0.2722   Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
   X
Mode:logical
NA's:569
```

```r
# Check the structure of the dataset
str(your_dataset)
```

```
'data.frame':   569 obs. of  33 variables:
 $ id                     : int  842302 842517 84300903 84348301 84358402 843786 844359
84458202 844981 84501001 ...
 $ diagnosis              : chr  "M" "M" "M" "M" ...
 $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
 $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
 $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
 $ area_mean              : num  1001 1326 1203 386 1297 ...
 $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
 $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
 $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
 $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
 $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
 $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
 $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
 $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
 $ area_worst             : num  2019 1956 1709 568 1575 ...
 $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
 $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
 $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
 $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
 $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
 $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
 $ X                      : logi  NA NA NA NA NA NA ...
```

```r
# Check for missing values
sum(is.na(your_dataset))
```

```
[1] 569
```

```r
# Descriptive statistics
describe <- your_dataset %>%
```

```
  summarise_all(list(mean = ~mean(., na.rm = TRUE), sd = ~sd(., na.rm = TRUE)))
print(describe)
```

```
  id_mean diagnosis_mean radius_mean_mean texture_mean_mean
1 30371831             NA         14.12729          19.28965
  perimeter_mean_mean area_mean_mean smoothness_mean_mean compactness_mean_mean
1            91.96903       654.8891           0.09636028              0.104341
  concavity_mean_mean concave.points_mean_mean symmetry_mean_mean
1          0.08879932               0.04891915          0.1811619
  fractal_dimension_mean_mean radius_se_mean texture_se_mean perimeter_se_mean
1                  0.06279761      0.4051721        1.216853          2.866059
  area_se_mean smoothness_se_mean compactness_se_mean concavity_se_mean
1     40.33708        0.007040979          0.02547814        0.03189372
  concave.points_se_mean symmetry_se_mean fractal_dimension_se_mean
1             0.01179614        0.0205423               0.003794904
  radius_worst_mean texture_worst_mean perimeter_worst_mean area_worst_mean
1          16.26919           25.67722             107.2612        880.5831
  smoothness_worst_mean compactness_worst_mean concavity_worst_mean
1             0.1323686               0.254265            0.2721885
  concave.points_worst_mean symmetry_worst_mean fractal_dimension_worst_mean
1                 0.1146062           0.2900756                    0.08394582
  X_mean     id_sd diagnosis_sd radius_mean_sd texture_mean_sd
1    NaN 125020586           NA       3.524049        4.301036
  perimeter_mean_sd area_mean_sd smoothness_mean_sd compactness_mean_sd
1          24.29898     351.9141         0.01406413          0.05281276
  concavity_mean_sd concave.points_mean_sd symmetry_mean_sd
1        0.07971981             0.03880284       0.02741428
  fractal_dimension_mean_sd radius_se_sd texture_se_sd perimeter_se_sd
1               0.007060363    0.2773127     0.5516484        2.021855
  area_se_sd smoothness_se_sd compactness_se_sd concavity_se_sd
1   45.49101      0.003002518        0.01790818      0.03018606
  concave.points_se_sd symmetry_se_sd fractal_dimension_se_sd radius_worst_sd
1          0.006170285    0.008266372             0.002646071        4.833242
  texture_worst_sd perimeter_worst_sd area_worst_sd smoothness_worst_sd
1         6.146258           33.60254       569.357          0.02283243
  compactness_worst_sd concavity_worst_sd concave.points_worst_sd
1            0.1573365          0.2086243              0.06573234
  symmetry_worst_sd fractal_dimension_worst_sd X_sd
1        0.06186747                 0.01806127   NA
```

```
num_rows <- nrow(your_dataset)
cat("Number of rows:", num_rows, "\n")
```

```
Number of rows: 569
```

```
num_cols <- ncol(your_dataset)
cat("Number of columns:", num_cols, "\n")
```

```
Number of columns: 33
```

```
unique_values <- sapply(your_dataset, function(x) length(unique(x)))
cat("Unique values per column:\n")
```

Unique values per column:

```
print(unique_values)
```

id	diagnosis	radius_mean
569	2	456
texture_mean	perimeter_mean	area_mean
479	522	539
smoothness_mean	compactness_mean	concavity_mean
474	537	537
concave.points_mean	symmetry_mean	fractal_dimension_mean
542	432	499
radius_se	texture_se	perimeter_se
540	519	533
area_se	smoothness_se	compactness_se
528	547	541
concavity_se	concave.points_se	symmetry_se
533	507	498
fractal_dimension_se	radius_worst	texture_worst
545	457	511
perimeter_worst	area_worst	smoothness_worst
514	544	411
compactness_worst	concavity_worst	concave.points_worst
529	539	492
symmetry_worst	fractal_dimension_worst	X
500	535	1

Questions

Question 1 : How does the accuracy of a models compare to another when predicting the diagnosis? Which model performs best on the test set?

Approach : - We will be using three models: SVM, Logistic Regression, Naive Baye's. - After training the model with the dataset, we will compare and analyze the the accuracy of the model's performance.

Why this question : We chose this question because comparing the accuracy of different models, like Logistic Regression, NB and SVM, is essential for identifying the best-performing model on the test set. This question helps you understand which model is most effective for predicting the diagnosis in a dataset.

Visuals : - We will plot a bar chart with model type on one axis and accuracy score on the other. This will show an excellent contrast between the models.

Question 2 : How do the Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores differ among the models? This can show which model offers the best trade-off between true positive and false positive rates.

Approach : We will be using three models: SVM ,LR and NB. - After training the model with the dataset, we will compare the ROC and the AUC which represents the probability of that the model, if given a randomly chosen positive and negative example, will rank the positive higher than the negative.

Why this question : This question is valuable because comparing ROC curves and AUC scores provides insight into each model's ability to balance true positives and false positives. It helps identify the model with the best overall performance, especially when handling imbalanced data or cases where both sensitivity and specificity are important.

Visuals : We will plot a 3D surface with true positive rate, false positive rate, and AUC values, where each model is represented as a unique surface. This gives a spatial view of each model's trade-off between TPR and FPR.

Question 3: How does tuning specific hyperparameters affect model performance?

Approach : - After the model training, we will use hyperparameter tuning to enhance the model's performance, optimization and then draw a comparison

Why this question : This question is important because tuning hyperparameters allows you to optimize each model's performance, potentially improving accuracy, precision, and recall. It helps you understand how specific parameter choices impact model effectiveness on this dataset. (Before and after hyper-parameter tuning.)

Visuals : We will be comparing accuracy after hyperparameter tuning with different models using a bar plot.

Question 4 : How do the top two correlated features with the target variable influence the classifications of SVM, Logistic Regression, and Naive Bayes?

Approach : Use the correlation matrix to find the two best features that have the highest correlation with the target variable. Train two of the best performing models on all the features and then visualize the classification regions and decision boundary using only the two best features mapped to the two axes while trained on all the features.

Why this question : This question allows us to understand how the top two influential features impact classification and provides insights into each model's decision-making process. It also highlights the effectiveness of the models in separating classes using key variables, even when trained on the full dataset.

Visuals : We will use a scatter plot with the top two features on each axis and visualize how the different models classify the data points using different colors to mark different classes.

Plan of Attack

Task Name	Status	Assignee	Due	Priority	Summary
Create Proposal	Completed	Gaurangi, Tushar	11/12/24	High	Finish proposal and upload to GitHub repo

Task Name	Status	Assignee	Due	Priority	Summary
Update the proposal after peer review	Completed	Gaurangi, Tushar,Viren, Bhaskar	11/18/24	high	Revise proposal according to the feedback
Question 1	WIP	Gaurangi, Tushar	11/19/2024	High	Finish Q1
Question 2	WIP	Viren Bhaskar	11/22/2024	High	Finish Q2
Question 3	WIP	Bhaskar, Tushar	11/23/2024	High	Finish Q3
Question 4	WIP	Viren, Gaurangi	11/27/2024	High	Finish Q4
Final analysis	WIP	Team	12/01/24	High	Finish design poster for ishowcase presentation
Poster review	WIP	Team	12/04/24	High	Do peer review in class
Final ishowcase presentation	WIP	Team	12/11/24	High	Do ishowcase presentation
Write-up	WIP	Team	12/13/24	High	Finish final write up

Final repository organisation

- **data/**: Contains dataset. It includes README, which details dataset parameters.
- **plots/**: Includes ggplot2 visualizations for each question. The README file explains each plot.
- **docs/**: Consist of write-up, final report. README outlines project goals and methods.
- **proposal.qmd**: Project proposal in qmd format.