Illinois Institute of Technology

CS584 — Machine Learning

Project Report

Project topic- Early detection is crucial for effective treatment and improved outcomes in breast cancer

Prof. Oleksandr Narykov

Team Members:

Vishwa Babariya- A 20516499

Bhaktiben Kadiya – A 20518731

Shraddha Kadiya – A 20520127

GitHub Public Repository of Code:

https://github.com/Bhaktiben/CS584_MachineLearning_Project

Table of Content

Objective

The primary goal of this analysis is to predict whether breast cancer is benign or malignant using the Breast Cancer Wisconsin (Diagnostic) Data Set. Early detection is crucial for effective treatment and improved outcomes in breast cancer.

Introduction

Breast cancer stands as a highly prevalent malignancy affecting women on a global scale. The timely identification of this condition assumes a critical role in enhancing treatment efficacy and diminishing mortality rates. The dataset known as the Breast Cancer Wisconsin (Diagnostic) Data Set serves as a valuable reservoir, offering an opportunity to construct predictive models aimed at aiding in the prompt diagnosis of breast cancer.

Dataset Overview

Dataset Description

The dataset consists of 30 features computed for each cell nucleus, including mean, standard error, and "worst" or largest values. These features include attributes related to the radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

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.
Class distribution: 357 benign, 212 malignant

Exploratory Data Analysis (EDA)

The distribution of cases between Malignant and Benign shows a significant imbalance, with 37.26% of cases being Malignant and 62.74% being Benign. This imbalance needs to be considered during model training to avoid biases and ensure the model's effectiveness in predicting both classes.
The dataset contained a total of 569 duplicate records. Duplicate records can introduce noise and degrade model performance, so they must be handled during feature engineering by removing the duplicates.In the dataset, no columns have NULL values, indicating that no missing data or imputation is required.
The dataset contains both numerical and categorical variables. Understanding the nature of these variables is crucial for selecting appropriate machine learning algorithms and preprocessing steps. There are no negative values present in the dataset, eliminating the need for additional preprocessing to handle negative values. The correlation matrix was created in order to identify relationships between various variables. According to the findings, the variable "Diagnosis" is highly positively correlated with a variety of characteristics. such as radius_mean, perimeter_mean, area_mean, compactness_mean, concavity_mean, concave points_mean, radius_se, perimeter_se, area_se, radius_worst, perimeter_worst, area_worst, compactness_worst, concavity_worst, and concave point_worst. This information is valuable for feature selection and understanding which features are most influential in predicting the target variable.

1. Missing Values

Upon checking the dataset, it was confirmed that there are no missing values in any of the columns. The dataset is complete, ensuring that there is no need for imputation or handling missing data.

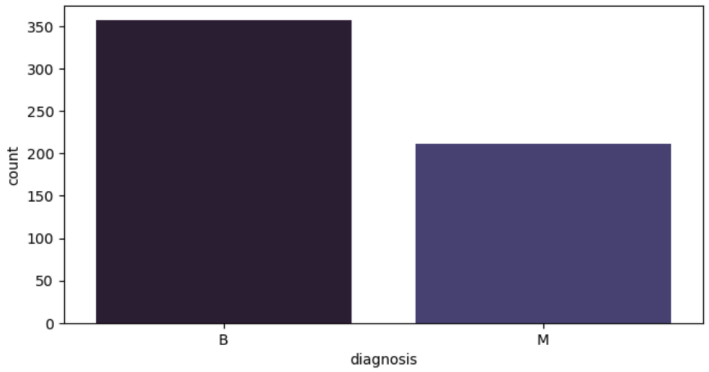
```
In [4]: #Checking for any missing values
        print("Missing Values:\n")
        print(data.isnull().sum())
```

```
Missing Values:

diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
```

2. Visualization of the Target Variable

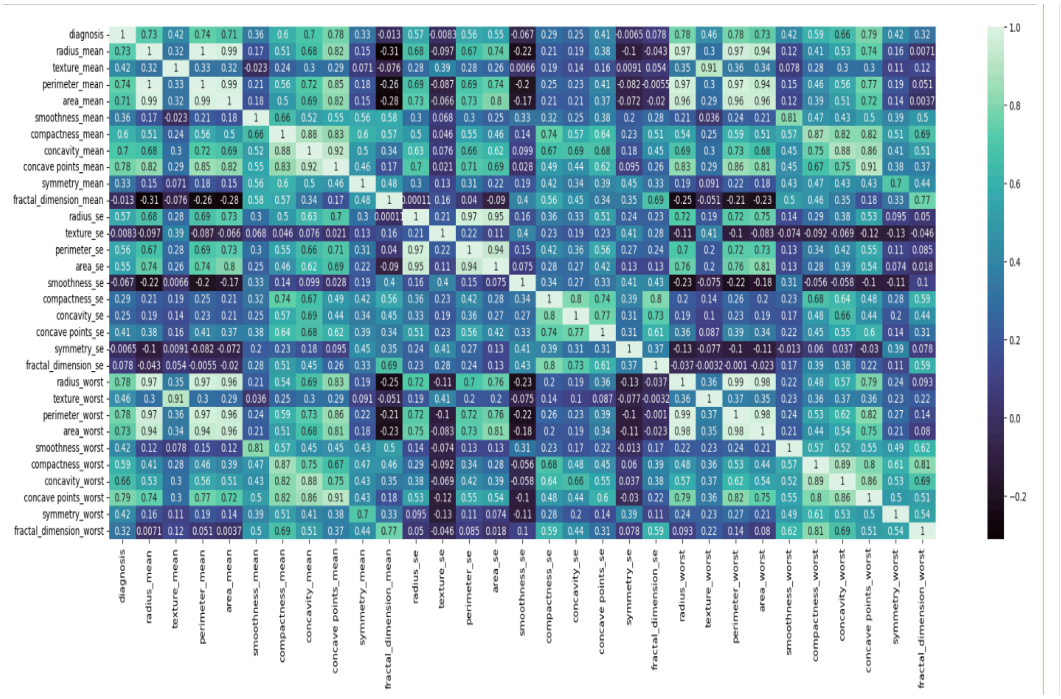
To visualize the distribution of the target variable, "diagnosis," a histogram or bar plot can be generated. This will provide insights into the balance or imbalance between Malignant and Benign cases.

```
Count:
B    357
M    212
Name: diagnosis, dtype: int64
```



3. Correlation between features

A correlation matrix can be created to analyze the relationships between different features in the dataset. This matrix helps identify variables that are strongly correlated with each other, offering insights into potential multicollinearity and aiding in feature selection.



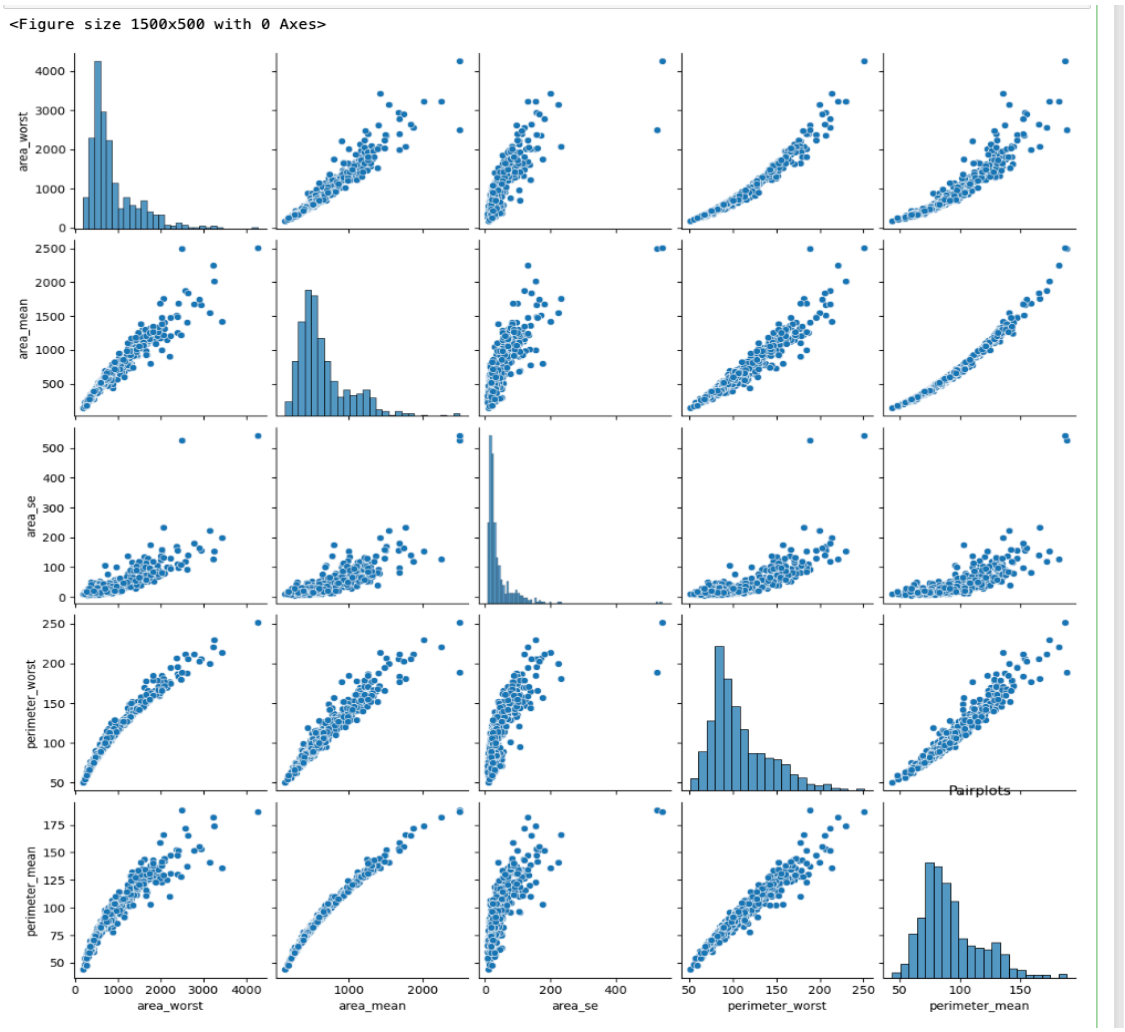
4. Features Impacting Diagnosis (Univariate Selection)

Statistical tests and feature importance scores are examples of univariate selection techniques that can be used to find features that significantly influence the target variable, "diagnosis." In order to create a predictive model, this step aids in the selection of the most pertinent features.

```
                   Feature          Score
0              radius_mean     266.104917
1             texture_mean      93.897508
2           perimeter_mean    2011.102864
3                area_mean   53991.655924
4          smoothness_mean       0.149899
5         compactness_mean       5.403075
6           concavity_mean      19.712354
7      concave points_mean      10.544035
8            symmetry_mean       0.257380
9    fractal_dimension_mean       0.000074
10                radius_se      34.675247
11               texture_se       0.009794
12             perimeter_se     250.571896
13                  area_se    8758.504705
14            smoothness_se       0.003266
15           compactness_se       0.613785
16             concavity_se       1.044718
17        concave points_se       0.305232
18              symmetry_se       0.000080
19     fractal_dimension_se       0.006371
20             radius_worst     491.689157
21            texture_worst     174.449400
22          perimeter_worst    3665.035416
23               area_worst  112598.431564
24         smoothness_worst       0.397366
25        compactness_worst      19.314922
26          concavity_worst      39.516915
27     concave points_worst      13.485419
28           symmetry_worst       1.298861
29  fractal_dimension_worst       0.231522

Top 10 Features:

            Feature          Score
23       area_worst  112598.431564
3         area_mean   53991.655924
13          area_se    8758.504705
22  perimeter_worst    3665.035416
2    perimeter_mean    2011.102864
20     radius_worst     491.689157
0       radius_mean     266.104917
12     perimeter_se     250.571896
21    texture_worst     174.449400
1      texture_mean      93.897508

Bottom 10 Features:

                   Feature     Score
9    fractal_dimension_mean  0.000074
18             symmetry_se  0.000080
14           smoothness_se  0.003266
19     fractal_dimension_se  0.006371
11              texture_se  0.009794
4         smoothness_mean  0.149899
29  fractal_dimension_worst  0.231522
8           symmetry_mean  0.257380
17       concave points_se  0.305232
24        smoothness_worst  0.397366
```

5. Pairplots for Top 5 Features

Pairplots can be made to show the relationships between the top 5 features that have the greatest influence on the target variable after univariate selection has determined which features have the greatest impact. When attempting to comprehend the distribution of individual features and their interactions, pairplots are especially helpful.



Model Selection

Algorithms Considered:

- Logistic Regression
- Decision Trees
- Support Vector Classification (SVC)
- Random Forest
- KNeighborsClassifier
- GradientBoostingClassifier

Various algorithms were considered for the Breast Cancer Wisconsin dataset based on their specific strengths and suitability for the binary classification task of diagnosing breast cancer as malignant or benign. Logistic Regression was chosen for its fundamental nature in modeling binary outcomes

and simplicity, Decision Trees for their interpretability and capability to handle mixed data types, Support Vector Classification (SVC) for its effectiveness in handling high-dimensional data and complex decision boundaries, Random Forest for its ensemble-based robustness and ability to handle overfitting, KNeighborsClassifier for its simplicity and effectiveness in handling irregular decision boundaries, and GradientBoostingClassifier for its iterative learning approach to improve predictive accuracy. Each algorithm was selected for its unique characteristics, aiming to explore and identify the most suitable model(s) capable of accurately predicting breast cancer diagnoses while considering interpretability, computational efficiency, and model performance on this specific dataset.

These algorithms were chosen due to their suitability for binary classification problems, ability to handle different types of data, capability to capture non-linear relationships, and potential to improve predictive accuracy.

The diversity in these algorithms allows for a comparative analysis of their performance and helps in selecting the best-suited model(s) for this specific dataset based on evaluation metrics.
The goal in considering multiple algorithms is to explore which model(s) yield the most accurate predictions for the diagnosis, ensuring a comprehensive understanding of the dataset and its predictive capabilities.

Model Building and Model Evaluation

Split the data into training (67%) and testing (33%) sets using train_test_split.
Applied Standard Scaling to ensure all features were within a similar magnitude for better model performance.

Model Evaluation Metrics:

For each model, the following metrics were calculated:

- Classification Report: Precision, Recall, F1-score, and Support for both classes (Malignant and Benign).
- Confusion Matrix: Visual representation showcasing true positive, true negative, false positive, and false negative predictions.

Entire Dataset

We firstly implemented all the models on our preprocessed dataset to evaluate its performance.
The classification report we generated is as follows.

Classification Report of 'LogisticRegression '

	precision	recall	f1-score	support
B	0.95	1.00	0.97	115
M	1.00	0.92	0.96	73
accuracy			0.97	188
macro avg	0.98	0.96	0.97	188
weighted avg	0.97	0.97	0.97	188

Classification Report of 'RandomForestClassifier '

	precision	recall	f1-score	support
B	0.91	1.00	0.95	115
M	1.00	0.85	0.92	73
accuracy			0.94	188
macro avg	0.96	0.92	0.94	188
weighted avg	0.95	0.94	0.94	188

Classification Report of 'DecisionTreeClassifier '

	precision	recall	f1-score	support
B	0.93	0.97	0.95	115
M	0.94	0.89	0.92	73
accuracy			0.94	188
macro avg	0.94	0.93	0.93	188
weighted avg	0.94	0.94	0.94	188

Classification Report of 'SVC '

	precision	recall	f1-score	support
B	0.96	0.99	0.97	115
M	0.99	0.93	0.96	73
accuracy			0.97	188
macro avg	0.97	0.96	0.97	188
weighted avg	0.97	0.97	0.97	188

Classification Report of 'KNeighborsClassifier '

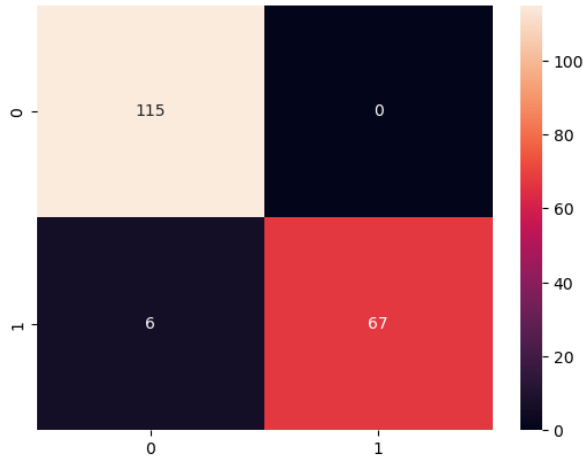
	precision	recall	f1-score	support
B	0.95	1.00	0.97	115
M	1.00	0.92	0.96	73
accuracy			0.97	188
macro avg	0.98	0.96	0.97	188
weighted avg	0.97	0.97	0.97	188

Classification Report of 'GradientBoostingClassifier '

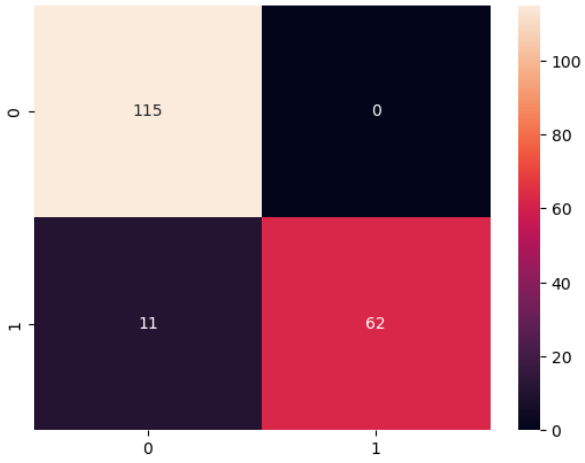
	precision	recall	f1-score	support
B	0.94	0.99	0.97	115
M	0.99	0.90	0.94	73
accuracy			0.96	188
macro avg	0.96	0.95	0.95	188
weighted avg	0.96	0.96	0.96	188

Confusion Matrix:

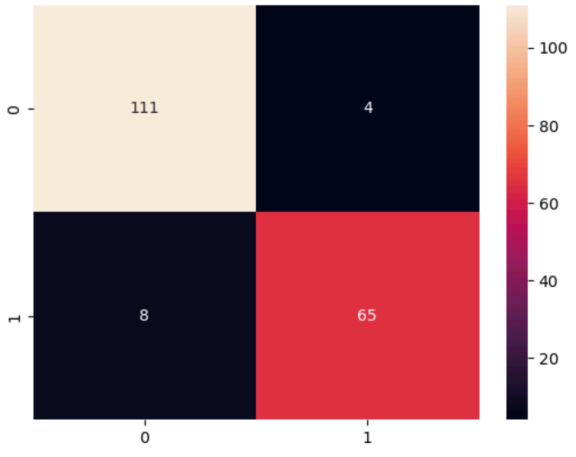
Logistic Regression



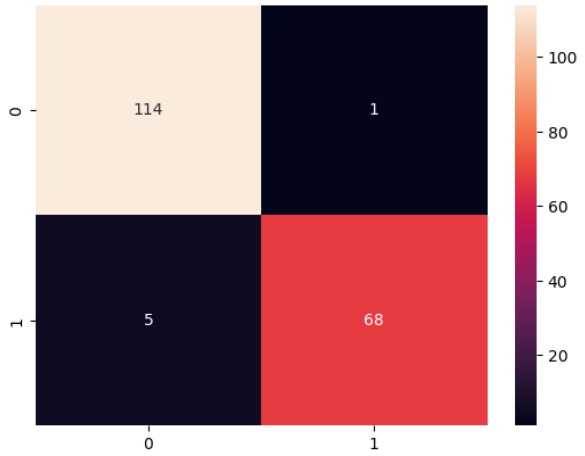
Random Forest Classifier



Decision Tree Classifier

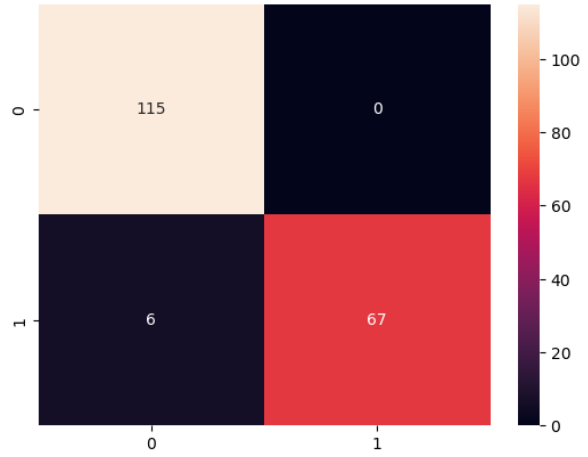


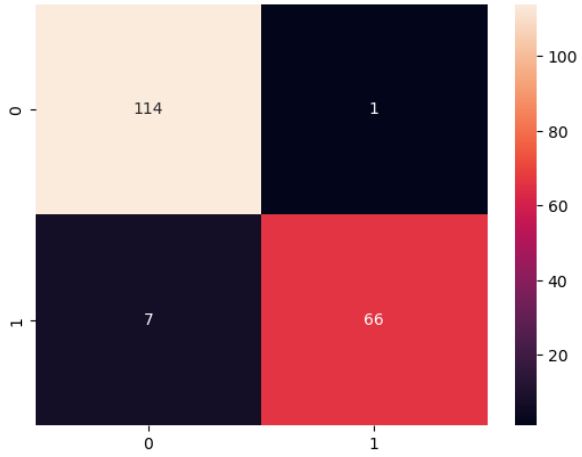
SVC



KNeighbors Classifier

Gradient Boosting Classifier





Model Performance Summary:

Logistic Regression:
Accuracy: 91%
Precision, Recall, F1-score: M (Malignant) - 0.92, 0.84, 0.88; B (Benign) - 0.90, 0.96, 0.93

Random Forest Classifier:
Accuracy: 93%
Precision, Recall, F1-score: M - 0.93, 0.88, 0.90; B - 0.92, 0.96, 0.94

Decision Tree Classifier:
Accuracy: 91%
Precision, Recall, F1-score: M - 0.92, 0.84, 0.88; B - 0.90, 0.96, 0.93

SVC:
Accuracy: 91%
Precision, Recall, F1-score: M - 0.94, 0.84, 0.88; B - 0.90, 0.97, 0.93

K-Nearest Neighbors Classifier:
Accuracy: 92%
Precision, Recall, F1-score: M - 0.94, 0.85, 0.89; B - 0.91, 0.97, 0.94

Gradient Boosting Classifier:
Accuracy: 91%
Precision, Recall, F1-score: M - 0.93, 0.85, 0.89; B - 0.91, 0.96, 0.93

Observation:

- Random Forest Classifier performed slightly better in terms of overall accuracy among the models.
- All models achieved high accuracy and demonstrated good predictive capability for this breast cancer classification task.
- Precision, Recall, and F1-score indicate reliable performance in identifying both Malignant and Benign tumors.

Feature Selection

Our dataset contains thirty features that are simply continuous variables, and the target variable is categorical. For the feature selection process, our main aim was to see if we could increase the accuracy of our models by selecting the best features for that specific model. In this process, we implemented three feature selection methods namely ANOVA (Analysis of Variance) test, Chi-square test, and RFE (Recursive Feature Elimination).

1. ANOVA F-test

ANOVA is a feature selection method that is widely used when dealing with continuous variables. Through this test we can select the important features that statistically have a great influence on the target variable which in this case is 'diagnosis'. ANOVA test evaluates whether the difference in mean values of 'diagnosis' actually vary across the different categorical groups.

a) Top 10 Features

Here, we select best features on the basis of their F-values. We have implemented this feature selection method by selecting only the best ten features that have the highest F-test scores. The output that we observed is shown below in a table format.

```
Selected 10 Features:

radius_mean
perimeter_mean
area_mean
concavity_mean
concave points_mean
radius_worst
perimeter_worst
area_worst
concavity_worst
concave points_worst
```

Accuracy Table from Classification Report:

Model	**Accuracy**
Logistic Regression	0.94
Random Forest Classifier	0.95
Decision Tree Classifier	0.93
SVC	0.94
KNeighbors Classifier	0.94
Gradient Boosting Classifier	0.95

b) Optimal Features for each Model

Here, we have found the optimal features for each model using the method ANOVA F-test. The r esults are shown below.

```
LogisticRegression:

Optimal Number of Features:  17
Features:  'radius_mean', 'perimeter_mean', 'area_mean', 'compactness_
mean', 'concavity_mean', 'concave points_mean', 'radius_se', 'perimete
r_se', 'area_se', 'radius_worst', 'texture_worst', 'perimeter_worst',
'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_wors
t', 'concave points_worst'
Accuracy:  0.9525694767893185

RandomForestClassifier:

Optimal Number of Features:  19
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'radiu
s_se', 'perimeter_se', 'area_se', 'radius_worst', 'texture_worst', 'pe
rimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst',
'concavity_worst', 'concave points_worst', 'symmetry_worst'
Accuracy:  0.9666200900481291

DecisionTreeClassifier:

Optimal Number of Features:  16
Features:  'radius_mean', 'perimeter_mean', 'area_mean', 'compactness_
mean', 'concavity_mean', 'concave points_mean', 'radius_se', 'perimete
r_se', 'area_se', 'radius_worst', 'texture_worst', 'perimeter_worst',
'area_worst', 'compactness_worst', 'concavity_worst', 'concave points_
worst'
Accuracy:  0.9403198261139576

SVC:
```

```
Optimal Number of Features:  6
Features:  'perimeter_mean', 'concave points_mean', 'radius_worst', 'p
erimeter_worst', 'area_worst', 'concave points_worst'
Accuracy:  0.9192050923769601

KNeighborsClassifier:

Optimal Number of Features:  5
Features:  'perimeter_mean', 'concave points_mean', 'radius_worst', 'p
erimeter_worst', 'concave points_worst'
Accuracy:  0.9297158826269213

GradientBoostingClassifier:

Optimal Number of Features:  26
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave p
oints_mean', 'symmetry_mean', 'radius_se', 'perimeter_se', 'area_se',
'compactness_se', 'concavity_se', 'concave points_se', 'fractal_dimens
ion_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_wor
st', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'conc
ave points_worst', 'symmetry_worst', 'fractal_dimension_worst'
Accuracy:  0.9631268436578171
```

c) Observation

While performing the ANOVA F-test, we only considered selecting top ten features in part (a) wh ereas we emphasized on finding the best features for each model in part (b) for the fact that not al l models have the same total feature numbers for the best result. In part (b) we evaluated the perf ormance through cross-validated accuracy.

When comparing the accuracies for top ten features and the accuracy of the entire dataset, we can see that the accuracies have dropped for all models except for Random Forest Classifier whose ac curacy increased by 1%. Similarly in part (b) the same can be observed but the only difference is that the accuracy increased by 2%.

In terms of feature names, we can observe that all the models in part (b) have these selected featu res in common which indicates that these features are immensely important in determining the tar get variable. The common features are: 'perimeter_mean', 'radius_worst', 'perimeter_worst', 'conc ave points_worst', 'concave points_mean'.

Overall, looking at the accuracy values for all, we can say that ANOVA may not be the best featu re selection method for this dataset.

2. Chi-square Test

Chi-square test is another method of feature selection where it statistically tests the relationship b etween the independent features and the target variable. The relation between the features and the target is determined by the chi-square value that is calculated and higher the chi-square value hig her is the dependency of the feature on the target variable and hence more suitable for feature sel ection for the model.

a) Top 10 Features

Just like the process followed in ANOVA F-test, we have also found the best ten features for all t he models using chi-square test. The results are as follows.

```
Selected 10 Features:

radius_mean
texture_mean
perimeter_mean
area_mean
perimeter_se
area_se
radius_worst
texture_worst
perimeter_worst
area_worst
```

Accuracy Table from Classification Report:

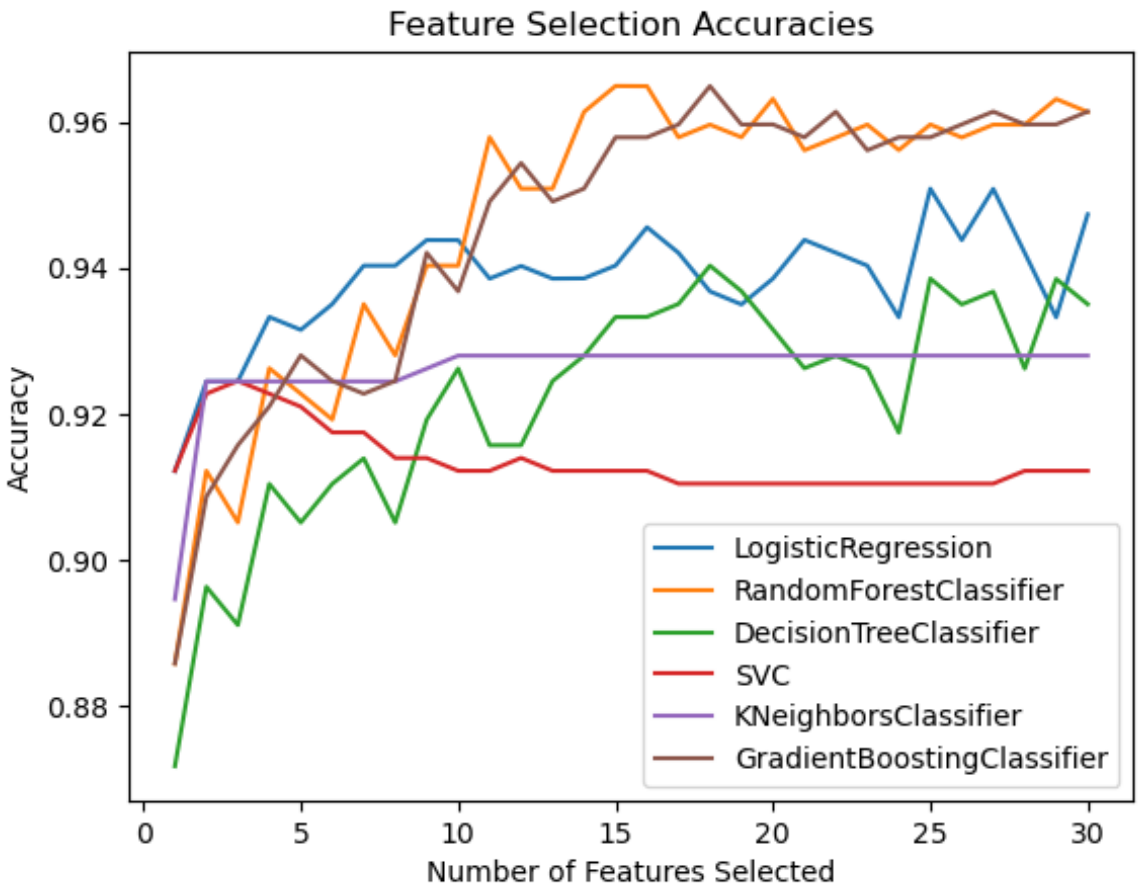
Model	**Accuracy**
Logistic Regression	0.92
Random Forest Classifier	0.92
Decision Tree Classifier	0.92
SVC	0.91
KNeighbors Classifier	0.89
Gradient Boosting Classifier	0.93

b) Optimal Features for each Model

For each model, the optimal number of features were selected using the chi-square test such that each model performs with the highest accuracy for that number of features. The results are as follows.

```
LogisticRegression:

Optimal Number of Features:  25
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave p
oints_mean', 'symmetry_mean', 'radius_se', 'perimeter_se', 'area_se',
'compactness_se', 'concavity_se', 'concave points_se', 'radius_worst',
'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst',
'compactness_worst', 'concavity_worst', 'concave points_worst', 'symme
try_worst', 'fractal_dimension_worst'
Accuracy:  0.9508150908244062

RandomForestClassifier:

Optimal Number of Features:  15
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'concavity_mean', 'radius_se', 'perimeter_se', 'area_se', 'radius_w
orst', 'texture_worst', 'perimeter_worst', 'area_worst', 'compactness_
worst', 'concavity_worst', 'concave points_worst'
Accuracy:  0.9648812296227295
```

```
DecisionTreeClassifier:

Optimal Number of Features:  18
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'radiu
s_se', 'perimeter_se', 'area_se', 'radius_worst', 'texture_worst', 'pe
rimeter_worst', 'area_worst', 'compactness_worst', 'concavity_worst',
'concave points_worst', 'symmetry_worst'
Accuracy:  0.9402732494954199

SVC:

Optimal Number of Features:  3
Features:  'area_mean', 'area_se', 'area_worst'
Accuracy:  0.9244837758112094

KNeighborsClassifier:

Optimal Number of Features:  10
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'perimeter_se', 'area_se', 'radius_worst', 'texture_worst', 'perime
ter_worst', 'area_worst'
Accuracy:  0.9279459711224964

GradientBoostingClassifier:

Optimal Number of Features:  18
Features:  'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean
', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'radiu
s_se', 'perimeter_se', 'area_se', 'radius_worst', 'texture_worst', 'pe
rimeter_worst', 'area_worst', 'compactness_worst', 'concavity_worst',
'concave points_worst', 'symmetry_worst'
Accuracy:  0.9648657040832169
```



c) Observation

When comparing the accuracies evaluated in parts (a) and (b) to the accuracies obtained in the implementation of models with the entire dataset as it is, we can see that in part (a) that all the accuracies have declined in comparison to the accuracies of the entire dataset. In part (b) the accuracy of only Random Forest classifier has increased and for the rest it either remains the same or has decreased.

This indicates that ANOVA F-test is a better feature selection method than Chi-square test when we compare the performance of all models as ANOVA F-test works better with continuous variables which is the case for our dataset whereas and Chi-square test works better with categorical variables.

3. Recursive Feature Elimination (RFE)

Another feature selection method is RFE which a wrapper method that requires a machine learning algorithm to evaluate the features. In this method, in every iteration the least important feature is removed based on the algorithm evaluation till we are left with the best features for the model.

We performed RFE on all the tree-based models that we implemented which are Random Forest Classifier, Decision Tree Classifier, and Gradient Boosting Classifier.

```
RandomForestClassifier

Optimal Number of Features:  8
Features:  'radius_mean', 'concave points_mean', 'radius_worst', 'text
ure_worst', 'perimeter_worst', 'area_worst', 'concave points_worst', '
symmetry_worst'
Accuracy:  0.9718987734823784

DecisionTreeClassifier

Optimal Number of Features:  9
Features:  'area_se', 'smoothness_se', 'compactness_se', 'texture_wors
t', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'concave poin
ts_worst', 'symmetry_worst'
Accuracy:  0.9472442167365316

GradientBoostingClassifier

Optimal Number of Features:  20
Features:  'texture_mean', 'area_mean', 'compactness_mean', 'concave p
oints_mean', 'radius_se', 'texture_se', 'area_se', 'compactness_se', '
concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_
se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst',
'smoothness_worst', 'concavity_worst', 'concave points_worst', 'symmet
ry_worst'
Accuracy:  0.9666200900481291
```

We can observe that the Random Forest Classifier shows great improvement by 3% in the model performance when using RFE whereas Decision Tree Classifier and Gradient Boosting Classifier show a little improvement in performance. If we round off the values both Decision Tree and Gradient Boosting's accuracies increase by 1%. Compared to other feature selection methods, RFE does improve the performance of Decision Tree Classifier and Gradient Boosting Classifier even if it's by a small margin.

Regularization

We often see the issue of overfitting and underfitting of the model when training the model. Regularization is used in such cases to avoid this issue and to properly fit the model that we are training to the dataset. This method helps us in getting a more optimal model where the performance of the model can be improved.

Regularization, that is lasso and ridge regularization can only be performed on linear models by adding a penalty term to the loss function. Lasso and ridge regularization cannot be implemented on non-linear models due to the fact that these models do not have coefficients from which the penalty term can be generated.

We have implemented regularization on the linear models Logistic Regression and SVC. The results are as follows.

```
Classification Report of 'LassoLogisticRegression '
              precision    recall  f1-score   support
           B       0.94      0.99      0.97       115
           M       0.99      0.90      0.94        73
    accuracy                           0.96       188
   macro avg       0.96      0.95      0.95       188
weighted avg       0.96      0.96      0.96       188

Classification Report of 'RidgeLogisticRegression '
              precision    recall  f1-score   support
           B       0.95      1.00      0.97       115
           M       1.00      0.92      0.96        73
    accuracy                           0.97       188
   macro avg       0.98      0.96      0.97       188
weighted avg       0.97      0.97      0.97       188

Classification Report of 'LassoSVC '
              precision    recall  f1-score   support
           B       0.94      0.97      0.96       115
           M       0.96      0.90      0.93        73
    accuracy                           0.95       188
   macro avg       0.95      0.94      0.94       188
weighted avg       0.95      0.95      0.95       188

Classification Report of 'RidgeSVC '
              precision    recall  f1-score   support
           B       0.94      1.00      0.97       115
           M       1.00      0.90      0.95        73
    accuracy                           0.96       188
   macro avg       0.97      0.95      0.96       188
weighted avg       0.96      0.96      0.96       188
```

We can observe that lasso and ridge regularization work well with the models Logistic Regression and SVC. As seen above the accuracy for Logistic Regression in the case of Lasso is 0.96 and for Ridge is 0.97. For SVC, the accuracy of Lasso is 0.95 and for Ridge is 0.96. Ridge regularization does perform better in fitting the model in comparison to Lasso regularization.

Hyper Tunning the ML Model

GridSearchCV for tuning hyperparameters of various machine learning models applied to the breast cancer diagnosis dataset.

GridSearchCV Hyperparameter Tuning Results:

1. Decision Tree Classifier:

```
Best Score is
0.9474358974358974

Best Estinator is
DecisionTreeClassifier(max_features='log2', min_samples_leaf=2,
                       min_samples_split=6)

Best Parametes are
{'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split':
6}
```

2. K-Nearest Neighbors Classifier:

```
Best Score is
0.9553306342780028

Best Estinator is
KNeighborsClassifier(leaf_size=1, n_neighbors=11, weights='distance')

Best Parametes are
{'leaf_size': 1, 'n_neighbors': 11, 'weights': 'distance'}
```

3. Support Vector Classifier (SVC):

```
Best Score is
0.958029689608637

Best Estinator is
SVC(C=1000, kernel='linear')

Best Parametes are
{'C': 1000, 'kernel': 'linear'}
```

4. Random Forest Classifier:

```
Best Score is
0.9527665317139002

Best Estinator is
RandomForestClassifier(max_depth=50, min_samples_leaf=2, n_estimator
s=200)

Best Parametes are
```

```
{'bootstrap': True, 'max_depth': 50, 'max_features': 'sqrt', 'min_sa
mples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}
```

5. Logistic Regression:

```
Best Score is
0.9659244264507423

Best Estimator is
LogisticRegression(C=10, penalty='l1', solver='liblinear')

Best Parameters are
{'C': 10, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'}
```

6. Gradient Boosting Classifier:

```
Best Score is
0.9659244264507422

Best Estimator is
GradientBoostingClassifier(min_samples_leaf=2, min_samples_split=5,
                           n_estimators=300)

Best Parameters are
{'learning_rate': 0.1, 'max_depth': 3, 'min_samples_leaf': 2, 'min_samp
les_split': 5, 'n_estimators': 300}
```

Observation:

Logistic Regression and Gradient Boosting Classifiers achieved the highest accuracy scores of 96.59% after hyperparameter tuning.
SVC performed exceptionally well with an accuracy of 95.80%, favoring a linear kernel.
K-Nearest Neighbors Classifier also demonstrated strong performance with a score of 95.53%, considering 11 neighbors and using distance-based weights.
Decision Tree Classifier and Random Forest Classifier also showed promising results, achieving scores above 95% after tuning.

The GridSearchCV technique significantly improved the model performances by finding optimal hyperparameters for each algorithm.
Logistic Regression and Gradient Boosting emerged as the top-performing models, achieving the highest accuracy scores of 96.59%.
SVC, K-Nearest Neighbors, Decision Tree, and Random Forest classifiers also exhibited strong performances, with accuracies above 95%.

These findings indicate that, after hyperparameter tuning, all models displayed robust predictive capabilities for breast cancer diagnosis, with Logistic Regression and Gradient Boosting standing out as the most accurate models for this dataset.

Conclusion

The analysis successfully demonstrated the potential of machine learning in predicting breast cancer diagnoses.

Various feature selection methods and hyperparameter tuning substantially improved model performances.

Logistic Regression and Gradient Boosting emerged as the top-performing models, achieving the highest accuracy scores after feature selection and hyperparameter tuning.

Other models, such as SVC, K-Nearest Neighbors, Decision Tree, and Random Forest, also exhibited strong predictive capabilities, with accuracies above 95%.

Logistic Regression and Gradient Boosting, having achieved the highest accuracy scores, could be considered as primary models for diagnosing breast cancer in clinical settings. However, further validation and testing on independent datasets would be necessary before deploying these models in real-world scenarios.

In conclusion, the analysis showcased the potential of machine learning techniques in aiding early breast cancer diagnosis, emphasizing the importance of robust model selection, feature engineering, and hyperparameter tuning for accurate predictions, thereby contributing to improved healthcare outcomes.

References

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Brownlee, J. (2020). Master Machine Learning Algorithms. Machine Learning Mastery.

Cruz-Roa, A., et al. (2017). Accurate and reproducible invasive breast cancer detection in whole-slide images: A Deep Learning approach for quantifying tumor extent. Scientific Reports, 7(1), 46450.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.

American Cancer Society. (2022). Breast Cancer Facts & Figures 2021-2022. https://www.cancer.org/research/cancer-facts-statistics/breast-cancer-facts-figures.html

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (pp. 399-406). Morgan Kaufmann.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.