Breast Cancer Prediction

Naga Sai Tejaswi Gandu
ID: 01987559
Department of Computer Science
Kennedy College of Sciences
University of Massachusetts Lowell
NagaSaiTejaswi_Gandu@student.uml.edu

Mihir Sriram
ID: 02008249
Department of Computer Science
Kennedy College of Sciences
University of Massachusetts Lowell
Mihir_Sriram@student.uml.edu

Balaji Revanth Guduru
ID: 02004098
Department of Computer Science
Kennedy College of Sciences
University of Massachusetts Lowell
BalajiRevanth_Revanth@student.uml.edu

Abstract - **There are numerous Machine Learning algorithms that can be used, depending on the situation, to handle classification problems in the field of data science. But because there are so many different algorithms accessible, there is constantly discussion about which algorithm to employ or which is the best machine learning algorithm for the classification task.**

We make use of the the UCI Breast Cancer Wisconsin (Diagnostic) Dataset to predict if a person has breast cancer or not based on various dependent properties of a person collected during a Census. We use different Classification algorithms like Logistic Regression, Support Vector Machine, Decision Tree, Multilayer Perceptron, XGBoost, and Random Forest algorithms to perform this task. Also, we analyze the outcomes of the methods mentioned using evaluation metrics such as Confusion Matrix, F1 score, Recall, Precision, and Accuracy in order to determine which algorithm is better appropriate for the given situation in order to make the most accurate prediction.

Keywords – Machine Learning, Classification, Logistic Regression, Support Vector Machine, Decision Tree Classifier, Multilayer Perceptron, XGBoost, Random Forest Classifier, Confusion Matrix, F1 score, Recall, Precision, Accuracy, Adult dataset

I. INTRODUCTION

Breast cancer is considered one of the most common cancers in women caused by various clinical, lifestyle, social, and economic factors. Machine learning has the potential to predict breast cancer based on features hidden in data.

In this report, we collect data from the UCI Breast Cancer Wisconsin (Diagnostic) Dataset, then preprocess and clean the data. The system is then trained to predict breast cancer using several methods.

Further, we study the features by correlating them and visualize the key features in the dataset. As the dataset consists of 30 features, we implemented PCA to reduce the dimensionality of the dataset into 6 principal components. We trained using algorithms like Logistic Regression, Support Vector Machine, Decision Tree, Multilayer Perceptron, XGBoost, and Random Forest to predict if a woman is diagnosed with Breast Cancer. Then we test the trained systems on the same test dataset and map the accuracy of the systems using the Confusion Matrix and Classification Report.

We use this problem as the basis for the problem of comparison of how different machine learning models perform on the same dataset. We compare the models using some evaluation metrics. We analyze the performance of each algorithm and come to a conclusion on which algorithm works best for the current dataset.

II. RELATED WORK

- Sanam Aamir, et.al [1] has implemented the prediction of breast cancer using supervised Machine Learning Techniques like Random Forest, SVM, and MLP.
- Mohammad Monirujjaman Khan, et. al. [2] used the Wisconsin Breast Cancer Diagnostic dataset and applied random forest, logistic regression, and decision tree to predict and found that the logistic regression model gave the best accuracy.
- Siham A. Mohammed, et. al. [3] used data mining algorithms to enhance the performance and increase the accuracy of Decision Trees, Naïve Bayes, and SMO algorithms.
- Krzysztof G [4] has implemented methods for heterogeneous forests of decision trees based on the Separability of Split Value.
- Rui Sarmento [5] has shown better options to choose the classifer to present the best accuracy.
- Rhamadina Fitrah Umami, et. al., [6] used the same dataset as we did, the UCO Breast Cancer Wisconsin dataset and have done the study to diagnose breast cancer using different machine learning algorithms and have achieved the highest of 99.4% for the Generalized Linear Model.

III. METHODOLOGY

A. The Dataset:

1) Dataset Description:

The data we use for this project is taken from the UCI Breast Cancer Wisconsin (Diagnostic) Dataset which is published by the Machine Learning Repository at the University of California, Irvine (UCI). This dataset comprises 569 instances and 32 attributes with no missing values. It has 1 categorical and the rest are continuous variables that comprise information on radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The diagnosis label in the data set, predicts whether the breast cancer is malignant or benign.

TABLE I. DATASET DESCRIPTION

Features	*Description*
id	ID number
diagnosis	The diagnosis of breast tissues (M = malignant, B = benign)
radius_mean	mean of distances from the center to points on the perimeter
texture_mean	The standard deviation of gray-scale values
perimeter_mean	mean size of the core tumor
area_mean	
smoothness_mean	mean of local variation in radius lengths
compactness_mean	mean of perimeter^2 / area - 1.0
concavity_mean	mean of the severity of concave portions of the contour
concave_points	mean for the number of concave portions of the contour
symmetry	
fractal dimension	"coastline approximation" - 1

B. Data Preprocessing:

1) Removing inconsistencies in data:

When the dataset was analyzed, we observed that there were no duplicates or missing data but that the column "Unnamed: 32" was not helpful for our prediction. So, we modified the dataset by dropping this column from the dataset.

C. Feature Study and Selection:

1) Feature-to-Feature Correlation Analysis:

When we performed correlation on the dataset, we observed that the radius_mean column has a correlation of 1 and 0.99 with perimeter_mean and area_mean columns, respectively. This was because the three columns essentially contained the same information, which is the physical size of the observation (the cell). For instance, the radius_mean column has a correlation of 0.97 with the radius_worst column. Also, there is multicollinearity between the attributes compactness, concavity, and concave points.

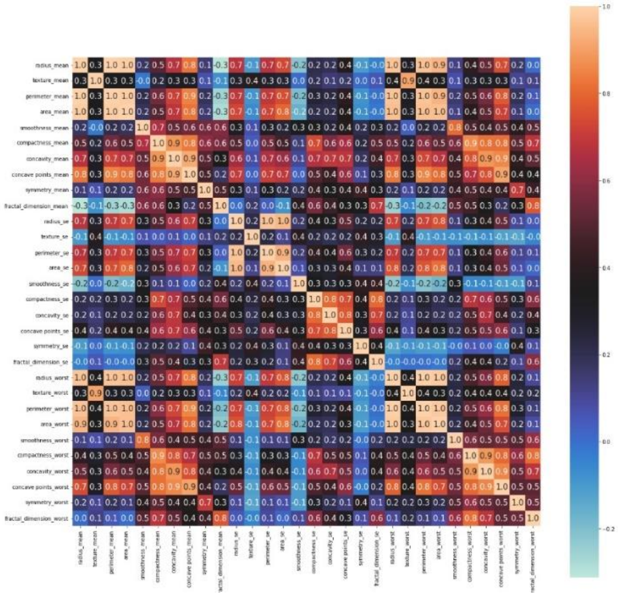


Fig 1. Heat-Map showing Feature-to-Feature and Feature-to-Label's Pearson Correlation Coefficients

D. Balancing the Dataset:

After observing the target label counts the data looks like imbalanced with B: 357, M: 212. So we implemented SMOTE technique using imblearn package.

SMOTE is a technique where when we have imbalanced data with more number in one label and very less in another label or vice versa it populates the data to make equal no of labels. SMOTE creates new minority instances by combining existing minority instances. It produces virtual training records for the minority class using linear interpolation. For each example in the minority class, these synthetic training records are constructed by randomly picking one or more of the k-nearest neighbors.

E. Dimensionality Reduction:

After doing the correlation analysis, we can remove specific columns which have high correlation but we tried using Principal Component Analysis (PCA) to identify the best features of its own. We used 20 principal components and plotted its variance. We observed that most of the variants are obtained using first six principal components, i.e., n_components = 5. Before passing the data to the PCA, the data should be normalized. So, we used a standard scalar from Sklearn to normalize the data.

We plotted its variance graph with number of components. We observed that 85% of the data spread is in first components in the Fig.3 so we choose n_components as 5

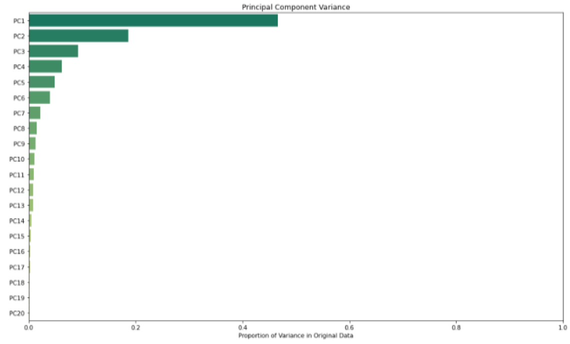


Fig 2. Variance Vs Principal Components

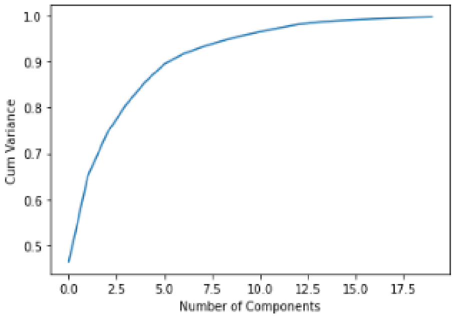


Fig 3. Variance Vs Number of Components

E. Preparing Train and Test Datasets:

To make sure that the Training Set and Testing Set had all of the unique attribute categories, the entire dataset is uniformly shuffled. Now that the dataset has been split into training and testing sets, 70% of the data is made available for training and the other 30% is used for testing.

F. Training the Models

1. Logistic Regression:

Logistic regression [7] is a type of statistical model that is used to predict binary outcomes from data. It is a supervised learning algorithm, which means that it is trained on labeled data, where the correct outcome (also known as the ground truth) is provided for each example in the training data. In logistic regression, the predicted outcome is a probability between 0 and 1, where 0 indicates that the example belongs to one class and 1 indicates that it belongs to the other class.

$$Sigmoid\ Function\ \hat{y}\ = g(z)\frac{1}{1+e^{-z}}$$

We use the loss function to assess the outcome after the model makes a prediction. We compute derivatives of the loss function with respect to their weights during this operation. We can use derivatives to determine how to change the weight and how much to lessen the model's loss.

$$Loss\ Function = -\frac{1}{m}\sum_{i=1}^{m} yi.\log(\hat{y}i) + (1 - yi).\log(1 - \hat{y}i)$$

2. Support Vector Machine Classifier:

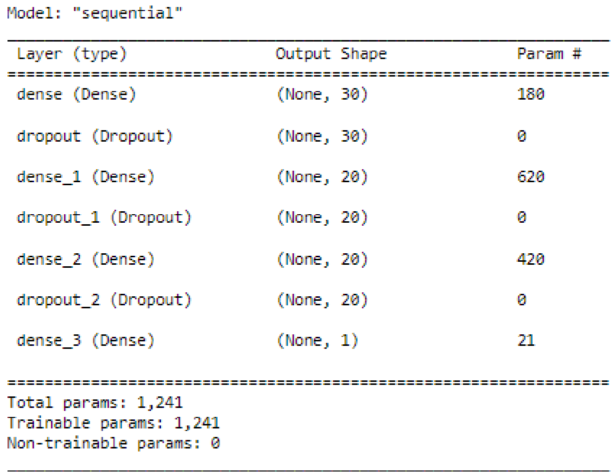
Support Vector Machines classify the data by locating a dividing line (or hyperplane) between two classes of data. The SVM algorithm uses data as input and, if it can, creates a line dividing the classes. To train the system for the present, we employ predefined techniques from the Python scikit-learn module. In this project, we used various SVM kernels to train multiple models using the support vector classifier algorithm. We used GridSearchCV to train the system with the hyperparameter values C:[1,10,100,1000], kernerl : ["poly," "rbf," , "sigmoid"] and gamma : [0.001,0.0001] and found the best paramaters as C=100 and gamma=0.001 The SVM Model is then trained using the best parameters and obtained the accuracy of 97.21%.We used this model to predict if a person has breast cancer.

3. Decision Tree Classifier:

To accomplish the classification on a dataset, Decision Trees employs the CART algorithm. The tree begins with one feature, then chooses the next feature to check based on its value, and so on until all the features have been used (or a specified number of features are used). Determining which feature should be chosen at each level is a crucial phase in the decision tree classifier. We employ the DecisionTreeClassifier method from the Python library scikit-learn for this. The Gini Index or Entropy of Information Gain on the dataset can be used to train a decision tree, or they can be used to determine which split is preferable. Scikit-DecisionTreeClassifier Learn's uses two input arrays made up of data features and target space as input before producing a full decision tree on the features. In this model we used GridsearchCv to find out the best hyper parameters for the following params criterion:[‘gini’,’entropy’], max_depth:range(1,10), min_samples_leaf:range(1,5), min_samples_split:range(2,10) and found the best hyper parametrs as criterion: entropy ,max_depth:6 min_samples_leaf : 1, min_samples_split:8 The model is trained using this tree, which is then used to forecast if a person has breast cancer.

4. Multilayer Perceptron Classifier:

MLP [8] is an acronym for Multilayer Perceptron, which is a type of feedforward artificial neural network. MLPs are composed of multiple layers of neurons, with each layer fully connected to the next. The input layer receives the input data and passes it on to the hidden layers, which use the input data to perform computations and generate output. The output from the hidden layers is then passed on to the output layer, which produces the final output of the network. MLPs [9] are commonly used for super vised learning tasks, such as classification and regression. They are able to model complex, non-linear relationships between the input data and the output. However, they can also be susceptible to overfitting if they are not properly regularized. A person's likelihood of having breast cancer is predicted using this model. The summary of our MLP model is in the fig 4



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 30)                180

 dropout (Dropout)           (None, 30)                0

 dense_1 (Dense)             (None, 20)                620

 dropout_1 (Dropout)         (None, 20)                0

 dense_2 (Dense)             (None, 20)                420

 dropout_2 (Dropout)         (None, 20)                0

 dense_3 (Dense)             (None, 1)                 21

=================================================================
Total params: 1,241
Trainable params: 1,241
Non-trainable params: 0
_________________________________________________________________
```

Fig 4. Multi Layer Perceptron model Summary

5. XGBoost Classifier:

XGBoost is an implementation of the gradient boosting algorithm, which is a specific type of ensemble learning method. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Here are a few key equations used in XGBoost: Objective function: The goal of gradient boosting is to find the function that minimizes the objective function. In XGBoost, the objective function is defined as the sum of the loss function evaluated over all training examples, plus a regularization term.

Loss function: The loss function measures the discrepancy between the predicted value and the true value for a single training example. XGBoost supports various loss functions for regression and classification tasks. For example, the squared error loss is commonly used for regression tasks, while the log loss is commonly used for classification tasks.

Regularization term: The regularization term helps prevent overfitting by penalizing models with excessive complexity. In XGBoost, the regularization term is defined as the sum of the square of the weights of all the decision trees in the model. This model is used to predict whether a person has breast cancer.
We used GridSearchCv to find out best hyperparametrs for the following parameters max_depth:[2,6,12] , learning rate:[0.3,0.1,0.03,0.05] and n_estimators:range(60,100,20). We found out the best hyperparameters are learning_rate:0.3,max_depth:2 and n_estimators:60

6. Random Forest Classifier:

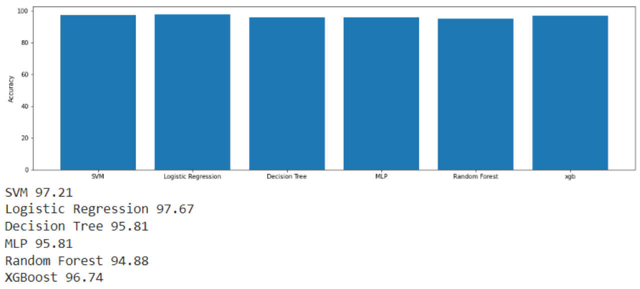
The Random Forest Classifier [10] employs the ensemble method, where each tree is a decision tree constructed from a sample of data with replacement using training data. The optimum split for these trees is found using all input features or a random collection of features taken from features in max features. The selection of these data samples is random. Overfitting and high variance are issues with decision trees that can be fixed by using Random Forests. We make use of the RandomForestClassifier algorithm from the Python sci-kit-learn module in our model. Similar to what we did with the Decision Tree, we used GridSearchCV to find out best hyperparameters for the following parameters n_estimators:[4,6,9,10,15],max_features:[‘log2’,’sqrt’,’auto’],criterion:[‘entropy’,’gini’],max_depth:[2,3,5,10],min_samples_split:[2,3,5],min_samples_leaf:[1,5,8] and found out that with criterion:entropy, max_depth:10,max_features:”log2”,n_estimators=15 are the best hypermarameter values.To predict whether a person has breast cancer, this model is used.

IV. RESULTS – COMPARISON OF MODELS

Using the training dataset, we have trained the Logistic Regression, Support Vector Machine, Decision Tree, Multilayer Perceptron, XGBoost, and Random Forest classifier algorithms in accordance with the above methods. To determine the outcome, or whether a person has breast cancer or not, we applied the trained models to the data features in the test dataset. To determine how accurate the trained models are, the resultant prediction was compared to the corresponding target feature in the testing set. We use a few model evaluation metrics from the Python scikit-learn library to compare the models.

A. Accuracy Score

This function determines the subset of classes predicted for a sample that precisely matches the corresponding subset of classes in the testing set. For each algorithm, we determined Accuracy Scores, and we created the graph in Fig 5. We observe that Decision Tree and Multilayer Perceptron have equal accuracy scores and Logistic Regression gave the highest accuracy.

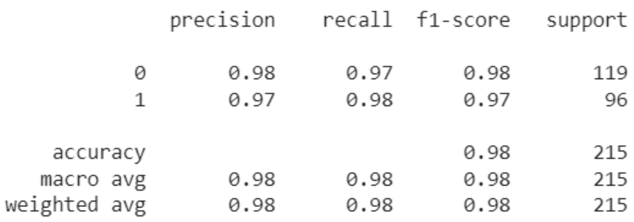


```
SVM 97.21
Logistic Regression 97.67
Decision Tree 95.81
MLP 95.81
Random Forest 94.88
XGBoost 96.74
```

Fig 5. Accuracy Scores of Algorithms

B. Classification Report

A classification model's classification Report provides a metrics value for the model broken down by class. The following are the trained model classification reports:

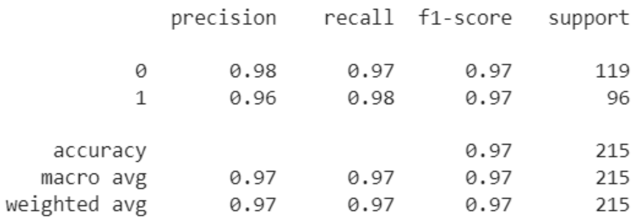


```
              precision    recall  f1-score   support

           0       0.98      0.97      0.98       119
           1       0.97      0.98      0.97        96

    accuracy                           0.98       215
   macro avg       0.98      0.98      0.98       215
weighted avg       0.98      0.98      0.98       215
```

Fig 6. Classification Report of Logistic Regression

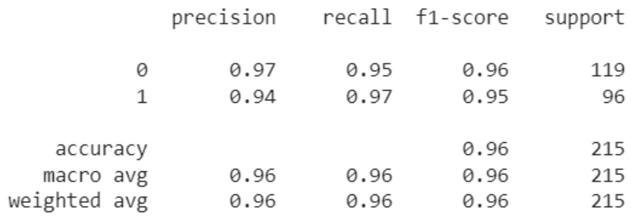


```
              precision    recall  f1-score   support

           0       0.98      0.97      0.97       119
           1       0.96      0.98      0.97        96

    accuracy                           0.97       215
   macro avg       0.97      0.97      0.97       215
weighted avg       0.97      0.97      0.97       215
```

Fig 7. Classification Report of Support Vector Machine Classifier

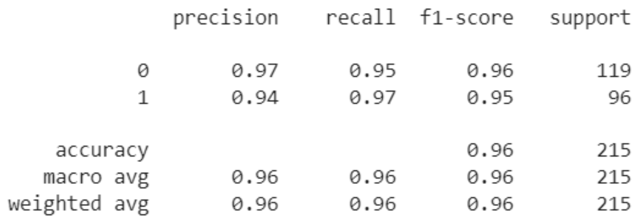


```
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       119
           1       0.94      0.97      0.95        96

    accuracy                           0.96       215
   macro avg       0.96      0.96      0.96       215
weighted avg       0.96      0.96      0.96       215
```

Fig 8. Classification Report of Decision Tree Classifier

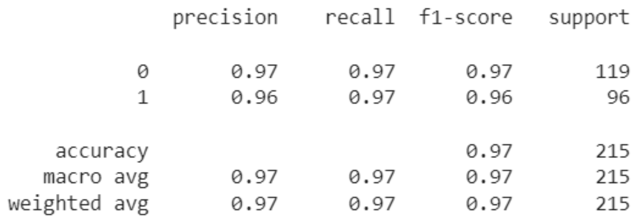


```
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       119
           1       0.94      0.97      0.95        96

    accuracy                           0.96       215
   macro avg       0.96      0.96      0.96       215
weighted avg       0.96      0.96      0.96       215
```

Fig 9. Classification Report of Multilayer Perceptron

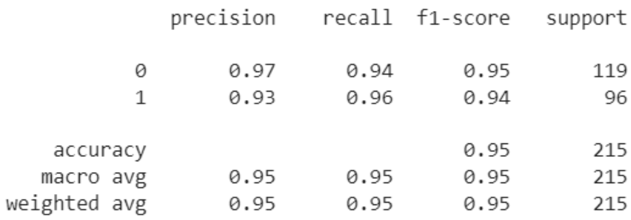


```
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       119
           1       0.96      0.97      0.96        96

    accuracy                           0.97       215
   macro avg       0.97      0.97      0.97       215
weighted avg       0.97      0.97      0.97       215
```

Fig 10. Classification Report of XGBoost Classifier



```
              precision    recall  f1-score   support

           0       0.97      0.94      0.95       119
           1       0.93      0.96      0.94        96

    accuracy                           0.95       215
   macro avg       0.95      0.95      0.95       215
weighted avg       0.95      0.95      0.95       215
```

Fig 11. Classification Report of Random Forest

C. Confusion Matrix:

A classification model's confusion matrix indicates how many True Positives, False Positives, False Negatives, and True Negatives the model predicts. For each model's confusion matrices, heatmaps have been plotted.

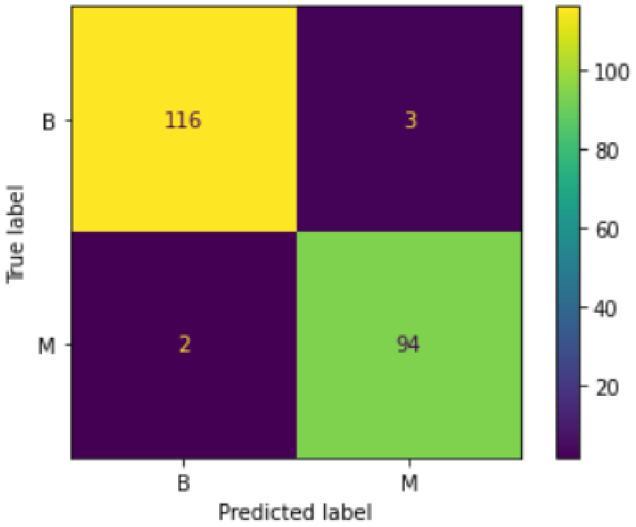


Fig 12. Confusion Matrix of Logistic Regression

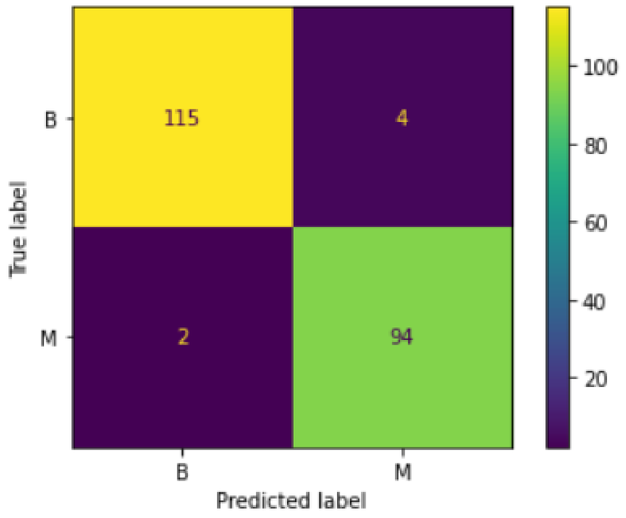


Fig 13. Confusion Matrix of Support Vector Machine

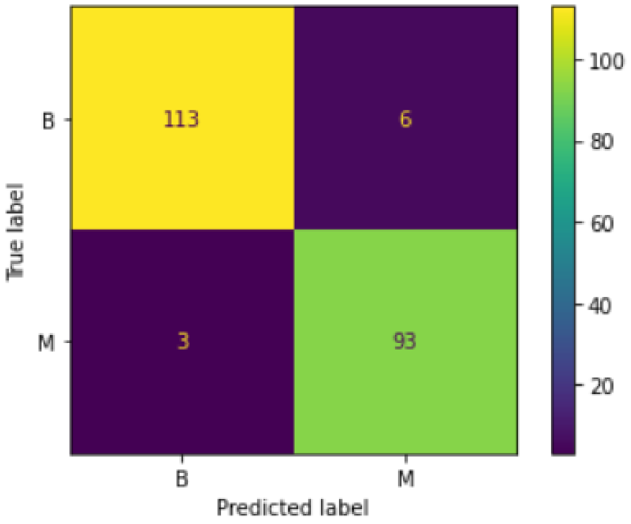


Fig 14. Confusion Matrix of Decision Tree

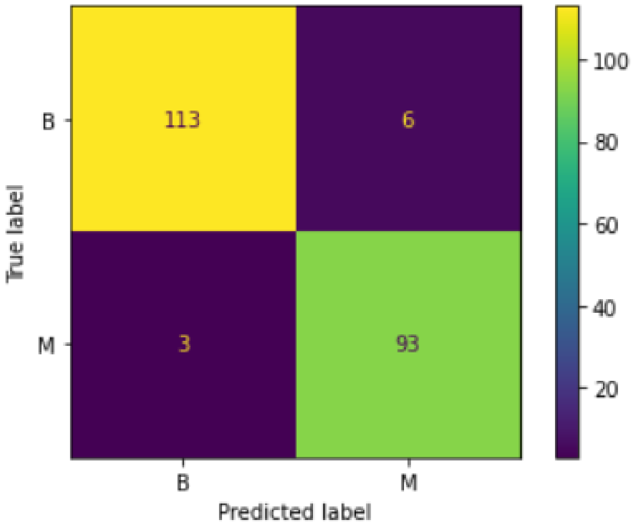


Fig 15. Confusion Matrix of Multilayer Perceptron

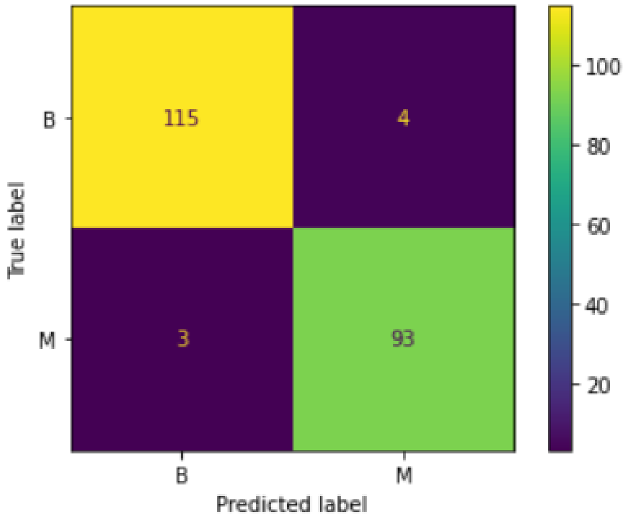


Fig 16. Confusion Matrix of XGBoost

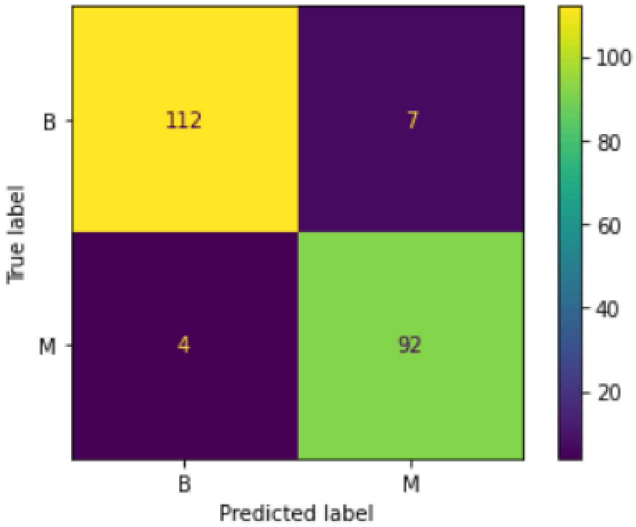


Fig 17. Confusion Matrix of Random Forest

D. Precision

The ratio of True Positives to All Positives is used to measure a model's precision. We first obtain the Classification Report for each algorithm before averaging the precision across both classes to determine the precision. And we plot a graph of precisions shown in Fig 18. We observe that the Logistic regression has the best Precision Score.

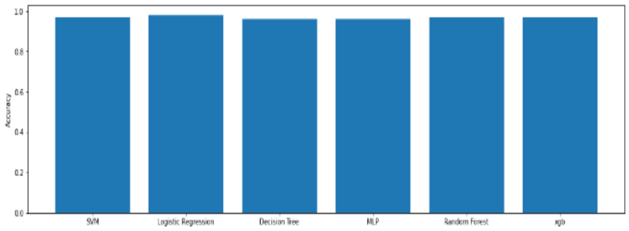


Fig 18. Precision of Algorithms

E. Recall

The measurement of how well our model properly identifies True Positives is known as recall of a model. We first obtain the Classification Report for each algorithm before averaging the recall across both classes to determine the recall. And we plot a graph of recalls shown in Fig 19. We observe that SVM and Logistic Regression have best score.

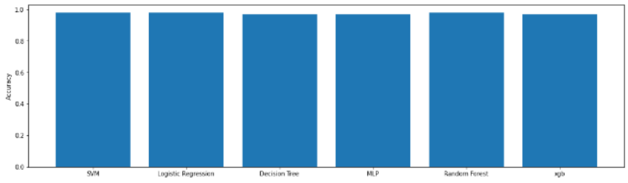


Fig 19. Recall of Algorithms

F. F1 Score

The Harmonic mean of the Precision and Recall is the F1-score. A statistic known as F1-Score denotes a high Precision and Recall score. We first obtain the Classification Report for each algorithm before averaging the F1-Scores of the two classes to determine the F1-Score. And we plot a graph of F1-Scores shown in Fig 20. We observe that SVM and Logistic Regression have best score.

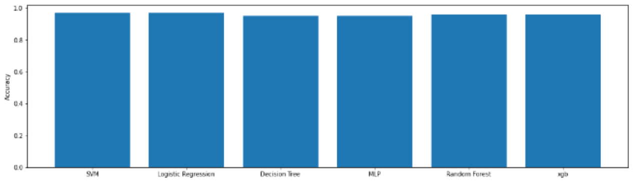


Fig 20. F1-Score of Algorithms

V. CONCLUSION

Observing the obtained results of the metric scores of all the trained algorithms we find that Logistic Regression and Support Vector Machine has the best Accuracy Scores; Logistic Regression have the best Precision on classes; Random Forest, SVM, and Logistic Regression have better Recall than other and when analyzed the F1-Score SVM and Logistic Regression have highest f1 score. Since the F1-Score is the Harmonic mean of Precision and Recall, we consider the F1-Score to be a more deciding factor in finding the most accurate model. And the data for each class in the dataset is imbalanced. Hence F1-Score is more feasible than Accuracy.

Hence, when comparing F1-Scores, SVM and Logistic Regression have the best scores, but when both algorithms are compared, we find Precision is better for the Logistic Regression and accuracy is also best for Logistic regression with a .40 difference with SVM, we conclude that Logistic Regression is the best possible model for predicting breast cancer.

REFERENCES

[1] Aamir, S., Rahim, A., Aamir, Z., Abbasi, S. F., Khan, M. S., Alhaisoni, M., Khan, M. A., Khan, K., & Ahmad, J. (2022, August 16). *Predicting breast cancer leveraging supervised machine learning techniques*. Computational and Mathematical Methods in Medicine. Retrieved December 12, 2022, from https://www.hindawi.com/journals/cmmm/2022/5869529/

[2] Monirujjaman Khan, M., Islam, S., Sarkar, S., Ayaz, F. I., Ananda, M. K., Tazin, T., Albraikan, A. A., & Almalki, F. A. (2022, April 11). *Machine learning based comparative analysis for breast cancer prediction*. Journal of Healthcare Engineering. Retrieved December 12, 2022, from https://www.hindawi.com/journals/jhe/2022/4365855/

[3] Mohammed, S. A., Darrab, S., Noaman, S. A., & Saake, G. (2020, July 11). *Analysis of breast cancer detection using different machine learning techniques*. Data Mining and Big Data: 5th International Conference, DMBD 2020, Belgrade, Serbia, July 14–20, 2020, Proceedings. Retrieved December 12, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7351679/

[4] Grabczewski, K., & Duch, W. (2002, August). *Heterogeneous forests of decision trees. - researchgate.net*. Heterogeneous Forests of Decision Trees. Retrieved December 12, 2022, from https://www.researchgate.net/profile/Duch-Wlodzislaw/publication/221078638_Heterogeneous_Forests_of_Decision_Trees/links/5ac3db160f7e9becc9d49318/Heterogeneous-Forests-of-Decision-Trees.pdf

[5] Sarmento, R. (2019, November). *Breast+Cancer+Wisconsin+(Diagnostic) - researchgate*. Breast Cancer Wisconsin (Diagnostic) Data Set. Retrieved December 12, 2022, from https://www.researchgate.net/profile/Rui-Sarmento-2/publication/337486825_Breast_Cancer_Wisconsin_Diagnostic_Data_Set/links/5ddba538458515dc2f4bcd7e/Breast-Cancer-Wisconsin-Diagnostic-Data-Set.pdf

[6] Fitrah Umami, R., & Sarno, R. (2020). Analysis of classification algorithm for Wisconsin Diagnosis Breast Cancer Data Study. *2020 International Seminar on Application for Technology of Information and Communication (ISemantic)*. https://doi.org/10.1109/isemantic50169.2020.9234295

[7] Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica*, 12–18. https://doi.org/10.11613/bm.2014.003

[8] POPESCU, M. A. R. I. U. S.-C. O. N. S. T. A. N. T. I. N., BALAS, V. A. L. E. N. T. I. N. A. E., POPESCU, L. I. L. I. A. N. A. P. E. R. E. S. C. U., & MASTORAKIS, N. I. K. O. S. (2009, July). *Multilayer Perceptron and neural networks - researchgate*. Retrieved December 12, 2022, from https://www.researchgate.net/profile/Nikos-Mastorakis/publication/228340819_Multilayer_perceptron_and_neural_networks/links/57ebb32208ae4119b2834599/Multilayer-perceptron-and-neural-networks.pdf

[9] Opitz, D., & Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, *11*, 169–198. https://doi.org/10.1613/jair.614

[10] Bahel, V., Pillai, S., & Malhotra, M. (2020). A comparative study on various binary classification algorithms and their improved variant for optimal performance. *2020 IEEE Region 10 Symposium (TENSYMP)*. https://doi.org/10.1109/tensymp50017.2020.9230877

APPENDIX

Project Contribution:

Logistic Regression Support Vector Machine	Naga Sai Tejaswi Gandu
Decision Tree Multilayer Perceptron	Mihir Sriram
XGBoost Random Forest	Balaji Revanth Guduru