Malaria Detection using Ensemble Classifier

In this project, we are trying to classify whether an image is a malaria infected image or not using ensemble classifiers.

Packages used:

1. scikit-learn
2. Numpy
3. Matplotlib
4. Seaborn

The classification problem was solved using the following steps:

1. Data Preprocessing and Cleaning
2. Split data into 3 chunks using stratified sampling
3. Handle skewness using smote algorithm in the training set
4. Reduce dimension of images using PCA
5. Apply ensemble method with three classifiers: adaboost algorithm, random forest classifier and svm
6. Appropriate hyperparameter selection (n_estimators in random forest and adaboost) using accuracy vs n_estimators plot
7. Evaluation of result on test dataset.

Data Preprocessing and Cleaning

At first, images were preprocessed before using the algorithms. Images were preprocessed by:
1. Converting to grayscale.
2. Applying Gaussian Blur.
3. Normalizing images to include values within [0-1] dividing every pixel by 255.
4. The images were standardized by subtracting the mean from each pixel and scaling to unit variance.
5. PCA was applied preserving 95% of variance in the data.

Split data into 3 chunks using stratified sampling

After data preprocessing and cleaning, the data were splitted into three chunks: train, validation and test containing 70%, 20% and 10% of total data respectively using stratified sampling. Thereafter, the data were shuffled in each of these sets.

Apply Ensemble Classifier

To build ensemble classifier, three algorithms were used:

1. AdaBoost Algorithm
2. Random Forest Classifier
3. Support Vector Machines

4. **Appropriate hyperparameter selection (n_estimators in random forest and adaboost) using accuracy vs n_estimators plot**

In case of both adaboost and random forest classifiers, we tried to estimate the appropriate value of n_estimators.

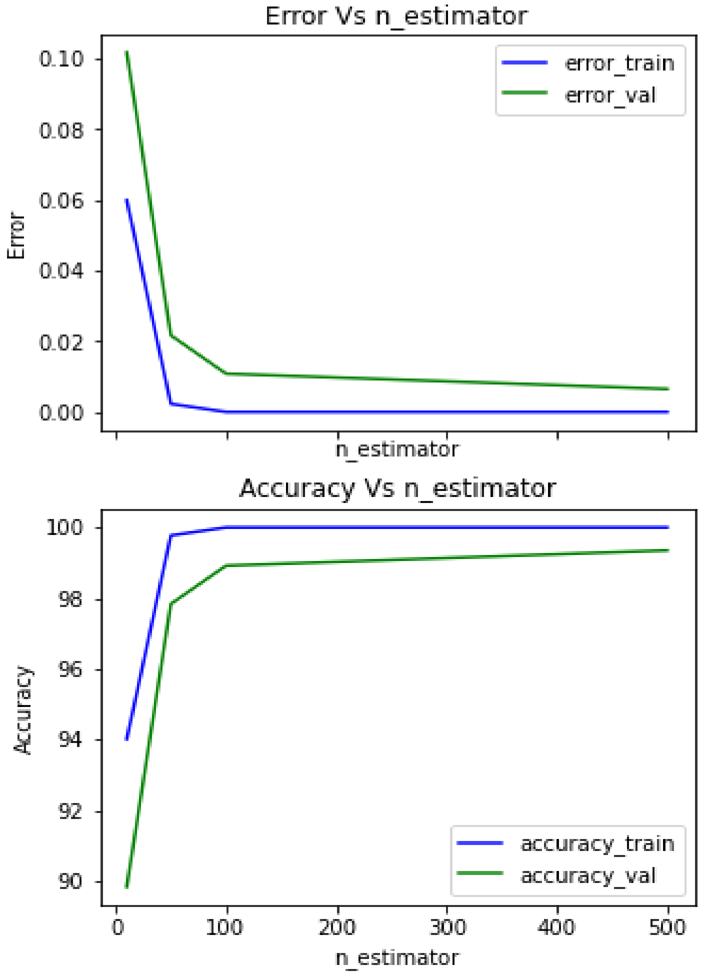


Figure 1. Error vs n_estimator and Accuracy vs n_estimator plot of Adaboost

From the validation plot, we chose n_estimator=100 as the appropriate number of estimators as the error was decreasing drastically until that point.

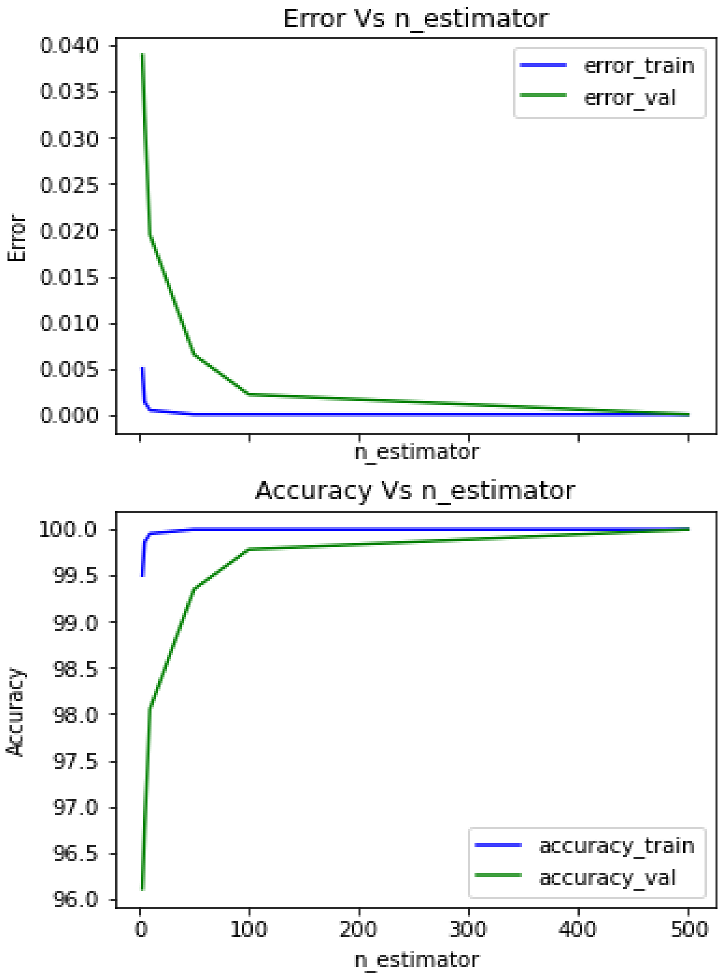


Figure 2. Error vs n_estimator and Accuracy vs n_estimator plot of Random Forest Classifier

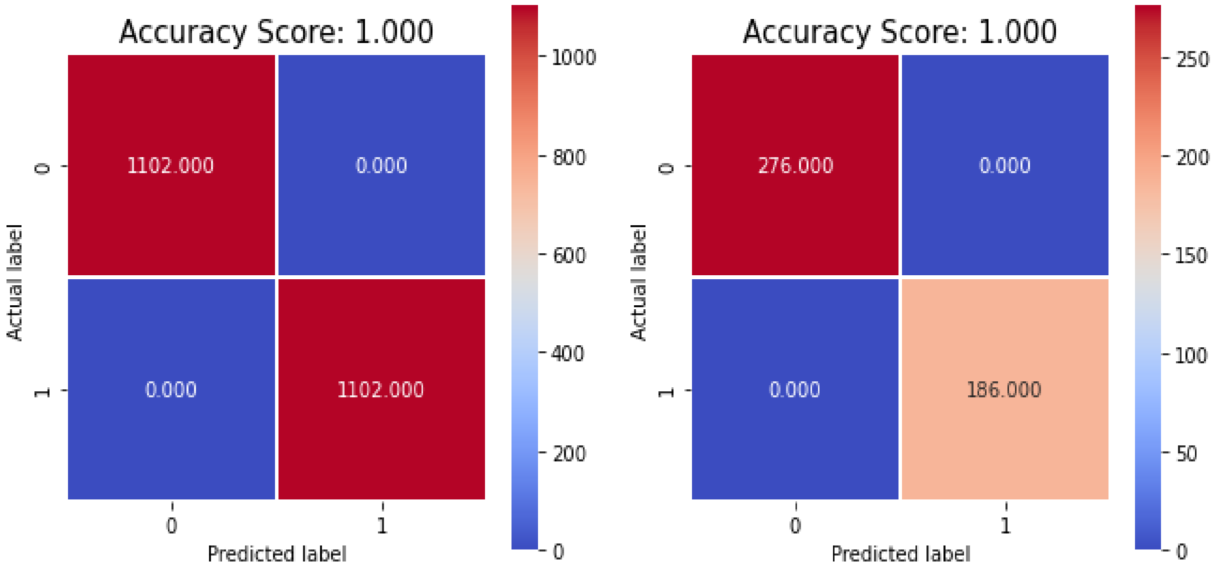
From the validation plot, we chose n_estimator=100 as the appropriate number of estimators for random forest too as the error was decreasing drastically until that point.

After selecting the appropriate value of n_estimator, we built an ensemble classifier with adaboost, random forest and svm.

Performance of training and validation set on the selected model

After selecting the proper model, we used it on predicting training and validation set images. The f1-score and accuracy on both training and validation set was 100%.

The confusion matrix obtained is given below:



(a) (b)

Figure 3. Confusion matrix of the results obtained from the prediction of training (a) and validation (b) dataset on the selected model

Plotting ROC AUC curve using training set and precision-recall curve using validation set:

As the training dataset was balanced after applying SMOTE oversampling algorithm and the test/validation dataset were imbalanced, so we used ROC-AUC curve for results of training set and Precision-Recall curve for validation and test set.

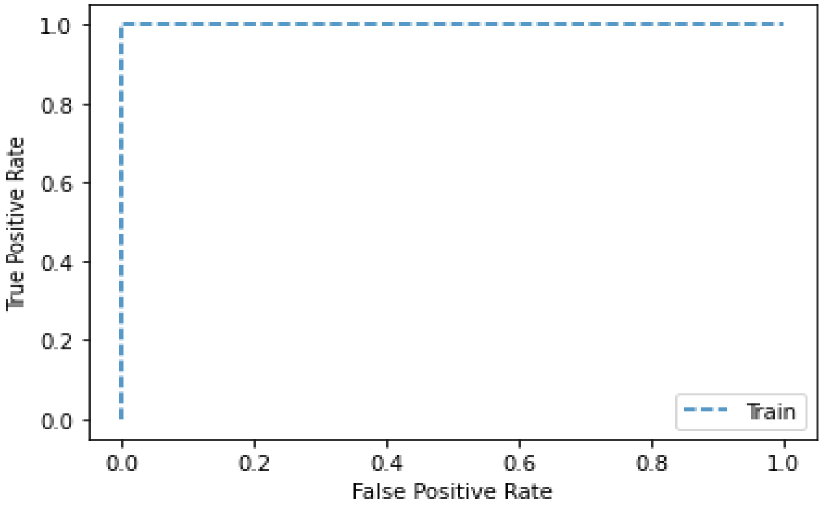


Figure 4: ROC-AUC curve (Training set)

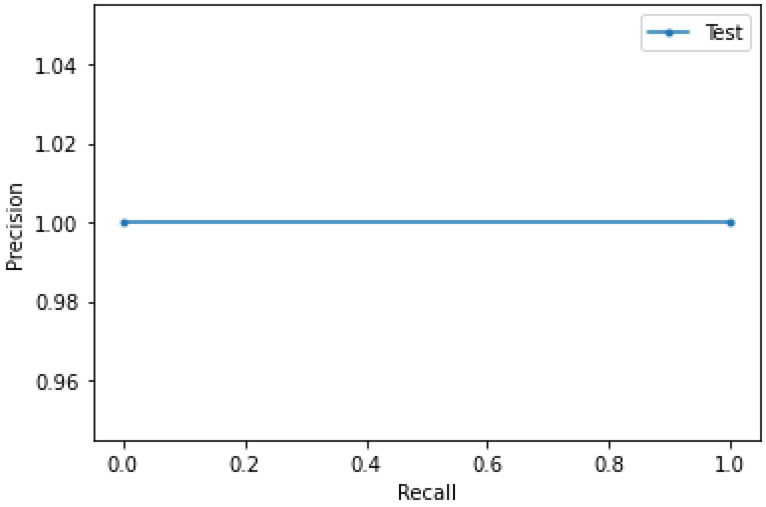


Figure 5: Precision-Recall Curve (Validation set)

Performance on test dataset

We also evaluated the performance of the selected model on the test set. Both f1-score and accuracy of the selected model was 100% on the test set as well. We obtained the following confusion matrix:

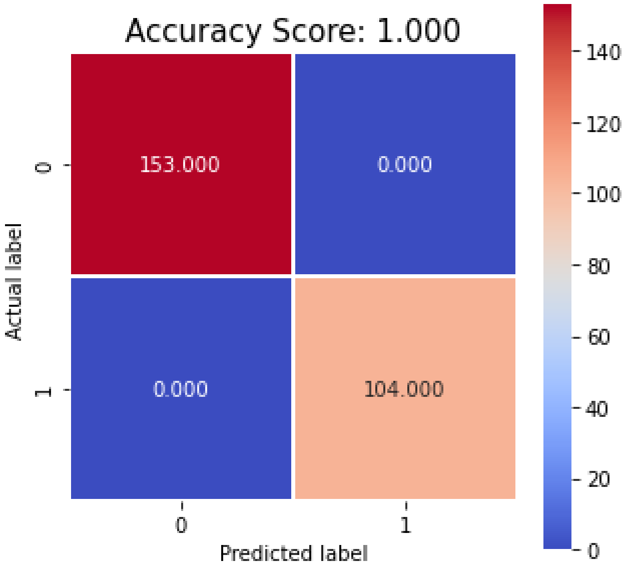


Figure 6: Confusion matrix to evaluate the performance on test set

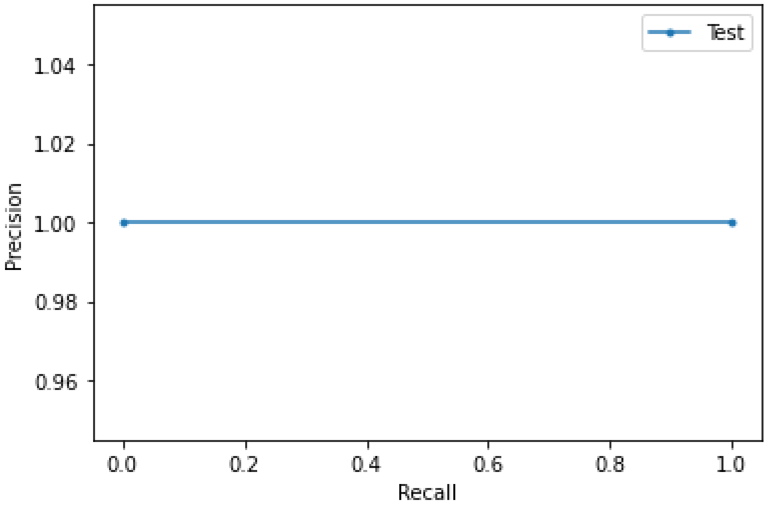


Figure 7: Precision Recall curve on the test set