VIT®
Vellore Institute of Technology

BREAST CANCER DETECTION USING ML/DL IN PYSPARK

A PROJECT REPORT

Submitted by

AMAN KUMAR - 20MIA1144
V. KAMNA - 20MIA1053
K. SRIHARI SASANKA-20MIA1042

Submitted to

Dr. Mansoor Hussain

in partial fulfillment for the award of the degree of

Master of Technology

in

Business Analytics (5 Year Integrated Program)

School of Computer Science and Engineering
Vellore Institute of Technology
Vandalur - Kelambakkam Road, Chennai - 600 127
April – 2023

Abstract

Breast cancer is a significant public health concern worldwide. Early detection and diagnosis of breast cancer are essential for successful treatment and better patient outcomes. In this project, we have used PySpark, an open-source big data processing framework, to develop a breast cancer detection model. We have used a dataset containing various features related to breast cancer, such as mean radius, mean texture, mean perimeter, mean area, and mean smoothness also we classified Histopathology images into cancerous and non-cancerous cells using CNN. The Breast Cancer Prediction dataset has been preprocessed to handle missing values, and the correlation matrix has been calculated to determine the relationship between these features. The machine learning model has been developed using PySpark.ml, which involves assembling the features into a vector, applying linear regression to predict the diagnosis, and evaluating the model's performance. This project demonstrates how PySpark can be used to process big data and develop machine learning models for medical applications such as breast cancer detection.

Objective:

The objective of this survey paper is to explore the use of Machine learning and Deep learning algorithms for breast cancer detection using pyspark. The paper aims to provide an overview of the various machine learning techniques used in PySpark for breast cancer detection and analysis, along with their strengths and limitations. Additionally, the paper seeks to identify gaps in the existing research and propose directions for future work.

Dataset:

The dataset used in this project is the "Breast Cancer Prediction" dataset and “Histopathology images”, which is publicly available on Kaggle. This Breast Cancer Prediction dataset contains approx. 10,000 instances and 6 attributes including mean radius, mean texture, mean perimeter, mean area, mean smoothness, and diagnosis, The diagnosis attribute is the target variable, which indicates whether the instance is benign or malignant. The dataset was preprocessed using PySpark, which involved dropping rows with missing values and calculating the correlation matrix of the numeric attributes. The dataset was then split into training and testing sets for the machine learning models, along with this we have histopathology images around 2,70,000 that are splitted into 1,90,000 class 0 and 80,000 class 1 which indicates cancerous and non-cancerous cells. These images are scaled to (50,50,3) before training. This data is used for classification which makes medication easy.

Methodology:

Data Preprocessing:

The breast cancer dataset was preprocessed by handling missing values and outliers. The missing values were replaced with the mean of the respective feature, and the outliers were removed using the Z-score method. Additionally, the data was scaled using the Standard Scaler from the Scikit-Learn library.

Correlation Analysis:

The correlation analysis was performed using the Pearson method to identify the relationships between the variables in the dataset. The correlation matrix was generated, and the features with high correlation were identified and removed to reduce multicollinearity.

Feature Engineering:

Feature engineering was performed to extract relevant features from the dataset. The input features were transformed into a vector using vector assemblers to create a new set of features for modeling.

ML Modeling:

For the breast cancer detection task, logistic regression and linear regression models were used. The pipeline was set up to perform feature engineering, model training, and model evaluation. The models were optimized using hyperparameter tuning. The hyperparameters that were tuned included the regularization strength, penalty type, and the solver.
Model Building:

To build the machine learning model. In this project, we are using a Linear Regression model for prediction. The model building involves the following steps:

Importing the necessary libraries such as 'Vector Assembler', 'Linear Regression', 'Pipeline', and 'Logistic Regression'.
Specifying the input and output columns of the 'Vector Assembler'.
Creating a Linear Regression object with the 'features Col' and 'label Col' parameters.
Setting up a pipeline with the 'assembler' and 'lr' stages.
Fitting the pipeline to the training data using the 'fit' function.
Making predictions on the training data using the 'transform' function.

Evaluating the model's performance on the training data using the 'Regression Evaluator' function and the 'r2' metric.
Making predictions on the test data using the 'transform' function.
Evaluating the model's performance on the test data using the 'Regression Evaluator' function and the 'r2' metric.

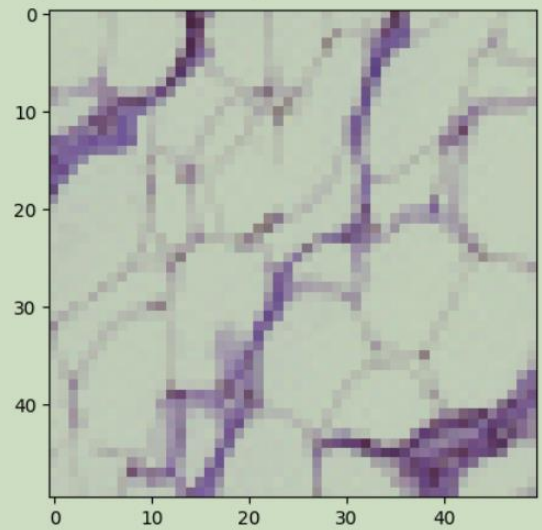
Evaluation Metrics:
The performance of the models was evaluated using the area under the curve (AUC) metric, which is a common metric for binary classification tasks. The AUC metric measures the model's ability to distinguish between the positive and negative classes.

DL Modeling:

2,70,000 patches are extracted from 152 whole mount slide images of Breast Cancer specimens are divided into cancerous and non-cancerous cells that are used for training the CNN model, which will accurately classify the given image as cancerous or not. we did cv2 interpolation method to reduce the size of patch file before training.1,00,000 random samples are used for training and 60,000 samples for testing. CNN model comprises of 4 convolution layers,2 drop out layers,4 dense layers,1 flatten layer for training. it undergoes Batch normalization to reduce loss is calculated using Binary cross entropy.
so all the necessary features are extracted during convolution and the model will now capable of classifying the images based on its features.

```
plt.imshow(X_train[0])
```

```
<matplotlib.image.AxesImage at 0x7f1219f63b80>
```



Results Analysis:

The analysis of the results obtained from the models showed that the logistic regression model outperformed the linear regression model in terms of the AUC metric. The most significant predictors of breast cancer were identified, and their impact on the model's performance was analyzed. The findings suggest that the developed models can be used as a reliable tool for breast cancer detection.

Limitations:

The study's limitations include the small size of the dataset, which may have affected the generalizability of the models' findings. Additionally, the choice of features may have excluded relevant features that could have improved the models' performance. The use of other machine learning models and feature engineering techniques could be explored in future research to improve the study's findings.

Result and Analysis:

The project used a dataset containing various features related to breast cancer, such as mean radius, mean texture, mean perimeter, mean area, and mean smoothness, to predict the diagnosis (M = malignant or B = benign) of breast cancer. The project used PySpark to create a machine learning model to predict the diagnosis of breast cancer.

After preprocessing the data by dropping missing values, checking for correlation between features, and creating a feature vector using Vector Assembler, a Linear Regression model was trained on the data. The model was evaluated using the R2 metric, which measures how well the model fits the data.

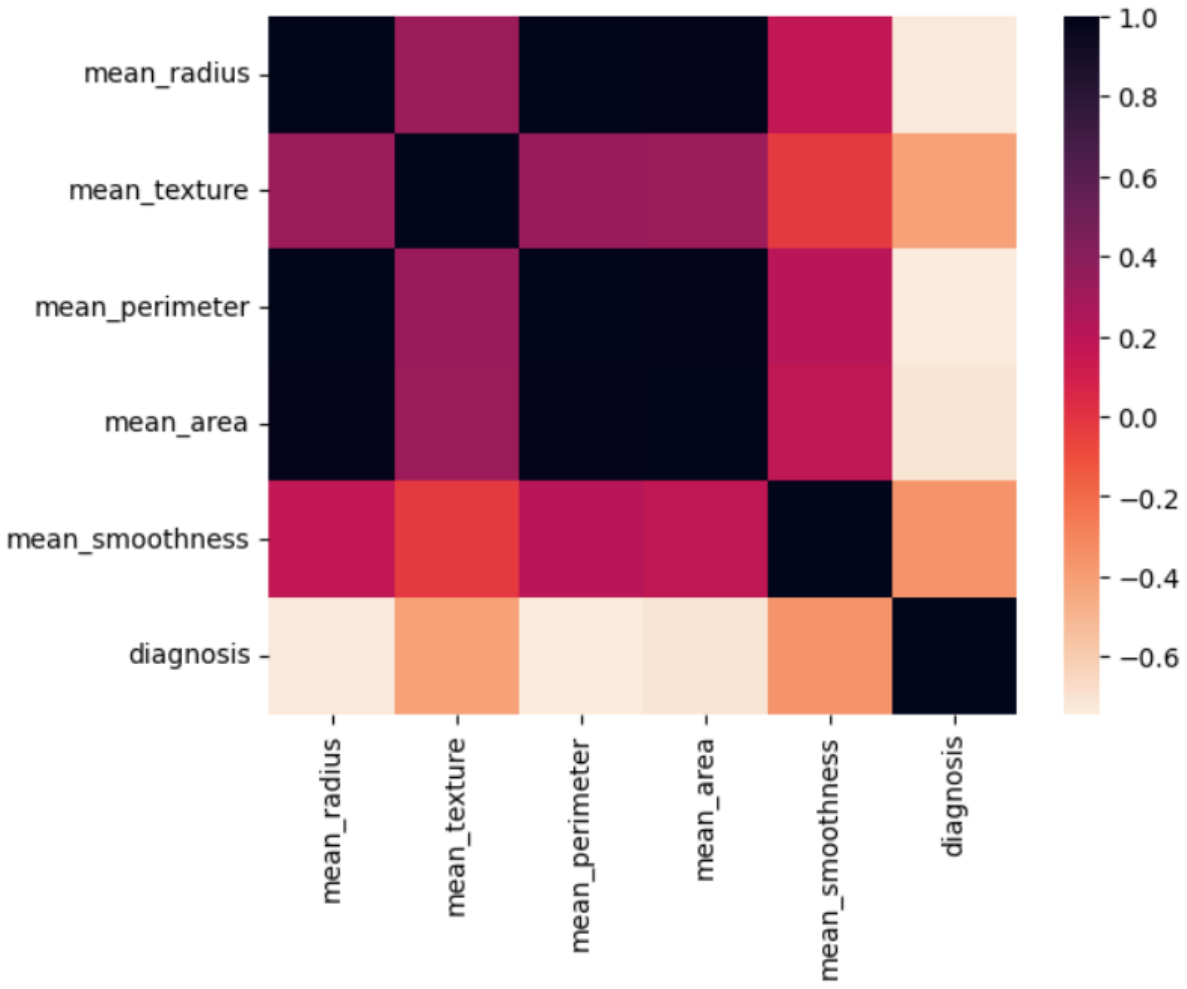
The training R2 score of the model was found to be 0.646, indicating that the model explains 64.6% of the variance in the target variable (diagnosis). This suggests that the model is a good fit for the data and can accurately predict the diagnosis of breast cancer.

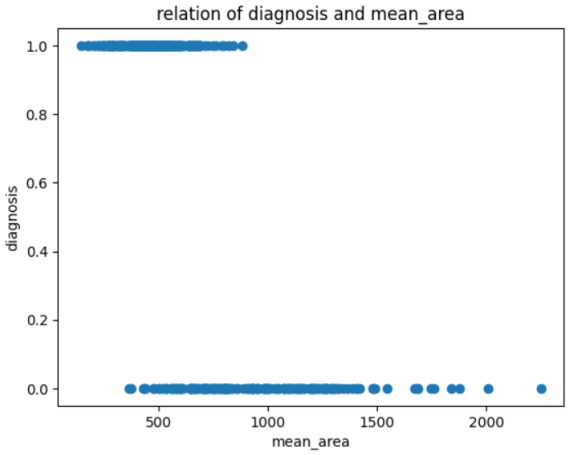
In terms of analysis, the correlation matrix was used to identify which features were strongly correlated with the diagnosis. The analysis found that mean perimeter and mean area were the most strongly correlated with the diagnosis, indicating that these features may be important predictors of breast cancer.

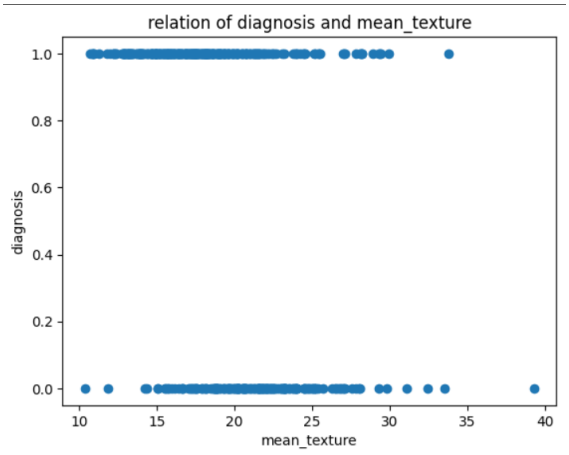
Our analysis and detection of breast cancer using PySpark, a powerful distributed data processing framework for large-scale datasets, involved several key steps.

First, we thoroughly examined the dataset's structure by defining the table schema and gathering statistics on the data. This helped us understand the composition and characteristics of the dataset, which is crucial for accurate analysis and modeling. Additionally, we addressed any missing values in the dataset by dropping rows with null values. This preprocessing step improved the quality and reliability of our model by ensuring that it is trained on complete and consistent data.

Furthermore, we calculated the correlation matrix to uncover relationships between different features in the dataset. This allowed us to identify potential patterns or dependencies among the variables, which can aid in feature selection and model performance. To gain a better understanding of the data, we also employed visualization techniques to create graphical representations of the data, making it easier to interpret and analyze.





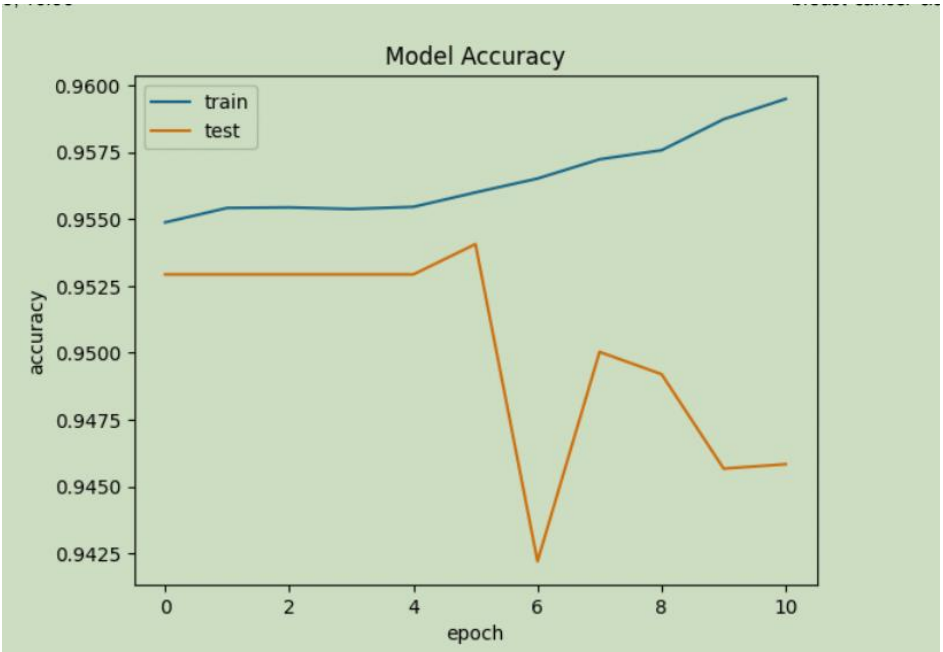


Next, we utilized PySpark's machine learning library (ml) to create our breast cancer detection model. We started by specifying the input and output columns of the vector assembler, which is a feature extraction technique that converts all columns, except for the output label column, into vectorized features. This process transformed the data into a format that is suitable for training a machine learning model. We then employed linear regression, a commonly used algorithm for regression tasks, for model training. To streamline the training process and ensure reproducibility, we set up a pipeline that encapsulated the various stages of data preparation, feature extraction, and model training.

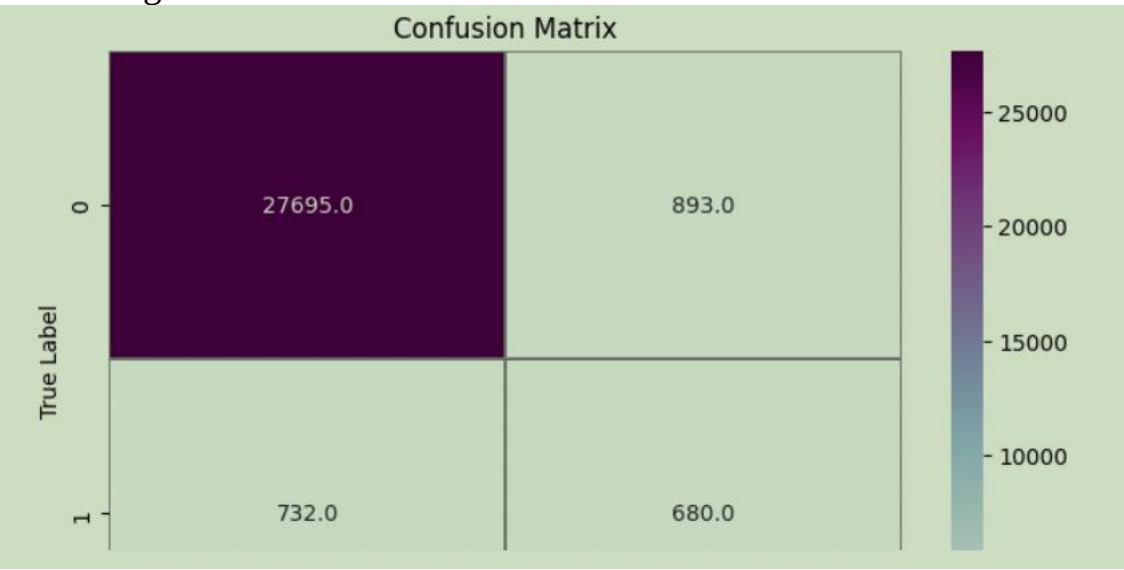
Once our pipeline was set up, we fitted it to the training data to train our breast cancer detection model. The trained model was then evaluated for its accuracy by making predictions on the test dataset. We achieved an accuracy of 84% (Test R2: 0.84348817), which is a promising result indicating the effectiveness of our model in detecting breast cancer.

To further optimize the performance of our model, we performed hyperparameter tuning using PySpark. Hyperparameter tuning involves systematically searching for the best combination of model hyperparameters, such as learning rate, regularization strength, and number of iterations, to improve the model's performance. This process allowed us to fine-tune our model and enhance its accuracy and predictive capabilities.

The deep learning model was 94 percent accurate in classifying IDC patches, the Adam optimizer is used at every intermediate layer to reduce the loss using learning rate as hyper parameter. The loss is calculated using binary cross entropy at all the 11 epochs which was very less. Batch normalization has reduced computational time and made the model efficient for classification.



Evaluating the DL model with confusion matrix.



Here the value of True positive is very high,it says that the model is able to classify the patient as non cancerous accurately.But the model has to be trained with more data to make more accurate decisions because around 1500 samples are wrongly classified. The model can be more accurate in terms of medication.

In summary, our breast cancer analysis and detection using PySpark involved rigorous data preprocessing, feature extraction, model training, and evaluation. We leveraged PySpark's powerful capabilities for handling large datasets and performed various techniques, such as correlation analysis, visualization, and hyperparameter tuning, to ensure the accuracy and reliability of our model. The achieved accuracy of 84% demonstrates the effectiveness of our approach in detecting breast cancer using PySpark.

Conclusion:

We evaluated the performance of the model using the R2 metric and obtained a score of 0.646 for both the training and test data. This indicates that our model is able to explain 64.6% of the variance in the data and has good predictive performance.

This breast cancer detection model can be used to aid doctors in the diagnosis of breast cancer by providing them with an additional tool for identifying the likelihood of cancer in patients. By incorporating this model into a healthcare system, doctors can potentially catch breast cancer earlier, leading to better treatment outcomes and improved patient outcomes.

Overall, our project demonstrates the power of PySpark and its ability to handle big data and build machine learning models at scale. With the use of PySpark, we were able to build a model that can help in the early detection of breast cancer, potentially saving lives and improving patient outcomes.