2023SP-COMP_SCI-5530-0002

Project Report: Breast Cancer Prediction

Using Machine Learning

Maruthi Phanindra Ayyagari 16323307

Mufrad Islam 16334246

Vamsi Kiran Murukutla 16326572

Ashok Karanam 16324075

Department of Computer Science

Submitted to: Yu Luo

Department of Computer Science

The University of Missouri, Kansas City, MO,64110.

Abstract

Breast cancer (BC) is one of the most common cancers among women worldwide, representing most new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem today.

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients from undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients in malignant or benign groups are the subject of much research. The goal is to classify whether the breast cancer is benign or malignant.

We will use the UCI Machine Learning Repository for the breast cancer dataset (http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29). The dataset we are going to use in this project is publicly available and was created by Dr. William H. Wolberg, a physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised Learning,Artificial Intelligence.

Introduction:

One of the most prevalent diseases in today's world is cancer among which breast cancer is caused by a variety of clinical, lifestyle, social, and economic variables. Breast cancer has become the most recurrent type of health issue among women especially for women in middle age. Early detection of breast cancer can help women cure this disease and the death rate can be reduced. The goal of this project is to predict breast cancer using machine learning algorithms.

Breast Cancer Classification:

Breast cancer classification is a supervised learning problem where the goal is to predict the diagnosis of breast cancer as either malignant or benign based on input features such as cell shape, size, and texture. It is a classification problem because the output variable is categorical, either malignant or benign. The dataset used in the Python program described earlier consists of several features related to breast cell characteristics, as well as the diagnosis of the cells as either malignant or benign.

Supervised learning is used because the dataset contains labeled data, which means that the output variable is known for each example in the dataset. In this case, the output variable is the diagnosis of breast cancer, and the input variables are the cell features.

The goal of the program is to train a machine learning model using the labeled dataset to predict the diagnosis of breast cancer for new, unlabeled examples. The program uses several machine learning algorithms to train and test the model and evaluate its performance using various metrics such as accuracy, precision, and recall.

Dataset: For this project, we have chosen the Breast Cancer Wisconsin Dataset of the following attributes and features.

The attributes are:

ID number

Diagnosis (M = Malignant, B = Benign)

Ten real-valued features are computed for each cell. They are:

Radius

Texture

Perimeter

Area

Smoothness

Compactness

Concavity

Concave points

Symmetry

Fractal Dimension

Dataset: "data.csv" is the dataset used in this program. It contains information about breast cancer diagnosis, including the characteristics of the cell nuclei present in the digitized image of a fine needle aspirate (FNA) of a breast mass. The dataset has 569 samples and 33 features. The targeted variables are the diagnosis column, which is either M (Malignant) or B (Benign).

Counting the number of Rows and Columns: Each row represents an individual patient, and the columns represent the features of each patient.

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471

842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017
843009 03	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279
843483 01	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052
843584 02	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043
843786	M	12.45	15.7	82.57	471.1	0.1278	0.17	0.1578	0.08089
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074
844582 02	M	13.71	20.83	90.2	579.9	0.1189	0.1645	0.09366	0.05985
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353
845010 01	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323
846100 02	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364
8466	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025

74 01									
84 79 90 02	M	14. 54	27.5 4	96.7 3	65 8.8	0.113 9	0.159 5	0.16 39	0.073 64
84 84 06	M	14. 68	20.1 3	94.7 4	68 4.5	0.098 67	0.072	0.07 395	0.052 59
84 86 20 01	M	16. 13	20.6 8	108. 1	79 8.8	0.117	0.202 2	0.17 22	0.102 8
84 90 14	M	19. 81	22.1 5	130	12 60	0.098 31	0.102 7	0.14 79	0.094 98
85 10 42 6	B	13. 54	14.3 6	87.4 6	56 6.3	0.097 79	0.081 29	0.06 664	0.047 81
85 10 65 3	B	13. 08	15.7 1	85.6 3	52 0	0.107 5	0.127	0.04 568	0.031 1
85 10 82 4	B	9.5 04	12.4 4	60.3 4	27 3.9	0.102 4	0.064 92	0.02 956	0.020 76
85 11 13 3	M	15. 34	14.2 6	102. 5	70 4.4	0.107 3	0.213 5	0.20 77	0.097 56
85 15 09	M	21. 16	23.0 4	137. 2	14 04	0.094 28	0.102 2	0.10 97	0.086 32
85 25 52	M	16. 65	21.3 8	110	90 4.6	0.112 1	0.145 7	0.15 25	0.091 7
85 26 31	M	17. 14	16.4	116	91 2.7	0.118 6	0.227 6	0.22 29	0.140 1
85 27 63	M	14. 58	21.5 3	97.4 1	64 4.8	0.105 4	0.186 8	0.14 25	0.087 83
85 27 81	M	18. 61	20.2 5	122. 1	10 94	0.094 4	0.106 6	0.14 9	0.077 31

85 29 73	M	15. 3	25.2 7	102. 4	73 2.4	0.108 2	0.169 7	0.16 83	0.087 51

Data Cleaning: Before properly using the data, we used the function called isna() to count the number of missing values in each column. We found a few columns with missing values and removed them using the dropna() function. We again checked the number of missing values using isna(), after removing the column with missing values. To make sure that no missing values remained.

Data Processing: To encode the categorical data in the diagnosis column, We then used the LabelEncoder() function of Scikit-Learn. To visualize the distribution of features and their correlation with each other, we also created a pair plot of the dataset using the pairplot() function of Seaborn. By using the corr() function of Pandas DataFrame the correlation of the columns in the dataset was obtained, and the correlation matrix was visualized using the heatmap() function of Seaborn [5].

Encoding the categorical data: Here, the values consisting in the “diagnosis” column are the categorical data. The values are Malignant (M) and Benign (B). We have encoded this value to the integer-type data

Creating a pair plot: We have created a pair plot for each row based on the data points of the “diagnosis” column. The following screenshot is provided for the first 5 indexes

Determining the correlation between the features of the dataset

The statistical relationship between two variables is referred to as their correlation. A correlation could be positive, meaning both variables move in the same direction. Negative, meaning that when one variable’s value increases, the other variable’s value decrease.

Visualizing the Correlation

A correlation heatmap is a graphical representation of a correlation matrix representing the correlation between different variables. Method **corr()** is invoked on the **Pandas DataFrame** to determine the correlation between different variables including predictor and response variables

Data normalization Techniques:

For the data normalization technique we used the StandardScaler() function of Scikit-Learn to scale the data. Scaling data ensures that each feature contributes equally to the analysis. Feature engineering or dimensionality reduction in this program is not performed.

train_test_split() function: The train_test_split() method is used to split our data into train and test sets. First, we need to divide our data into features (X) and labels (y). The dataframe gets divided into X_train, X_test, y_train, and y_test. X_train and y_train sets are used for training and fitting the model

Models Analysis: We have tested and trained our datasets using 3 different models.

Logistic Regression model: Logistic regression aims to solve classification problems. It does this by predicting categorical outcomes, unlike linear regression which predicts a continuous outcome.

Decision Tree Classifier: The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute.

Random Forest Classifier: One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

We have tested the accuracy of each model. The results are shown in the following screenshot:

A table that is used in classification problems to assess where errors in the model were made. The rows represent the actual classes the outcomes should have been. While the columns represent the predictions we have made

Visualization Analysis: We used the countplot() function of Seaborn to visualize the count of the two types of cells as shown above. We also created a pair plot of the dataset using the pairplot() function of Seaborn to visualize the distribution of features and their correlation with each other as shown above. The correlation matrix was visualized using the heatmap() function of Seaborn. These visualizations helped us to understand the distribution of data and the correlation between features.

Conclusion: The proposed machine-learning models could predict breast cancer as the early detection of this disease could help slow down the progress of the disease and reduce the mortalityrate through appropriate therapeutic interventions at the right time. Applying different machine learning approaches, accessibility to bigger datasets from different institutions (multi-center study)and considering key features from a variety of relevant data sources could improve the performance of modeling.

References

[1] "Intro to Machine Learning | Udacity." Intro to Machine Learning | Udacity. AccessedApril 27, 2016. https://www.udacity.com/course/intro-to-machine-learning--ud120.

[2] "Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition. Datasets:Coronary Heart Disease Dataset." Elements of Statistical Learning: DataMining, Inference, and Prediction. 2nd Edition. Accessed April 27, 2016. http://statweb.stanford.edu/~tibs/ElemStatLearn/.

[3] Schö lkopf, Bernhard, Christopher J. C. Burges, and Alexander J. Smola. Advances inKernel Methods: Support Vector Learning. Cambridge, MA: MIT Press, 1999.

[4] Norving, Peter, and Stuart Russel. Artificial Intelligence: A Modern Approach. S.l.:Pearson Education Limited, 2013.

[5] Witten, I. H., and Eibe Frank. Data Mining: Practical Machine Learning Tools andTechniques. Amsterdam: Morgan Kaufman, 2005.