Available online at: http://www.ijmtst.com/vol5issue11.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 05, Issue No: 11, November 2019

Steps to Classify the Breast Cancer type accuracy using Support Vector Machine – AI Machine Learning

Subash Kumar

Bachelor of Engineering specialized in Computer Science, Anna University, India.

To Cite this Article

Subash Kumar, "Steps to Classify the Breast Cancer type accuracy using Support Vector Machine – AI Machine Learning", *International Journal for Modern Trends in Science and Technology*, Vol. 05, Issue 11, November 2019, pp.-157-160.

Article Info

Received on 01-November-2019, Revised on 19-November-2019, Accepted on 21-November-2019, Published on 26-November-2019.

ABSTRACT

Timely diagnosis of any disease is critical in medical field, with increasing population of breast cancer patients, this paper is dedicated to all medical professionals who are trying to save many lives. Machine learning algorithm such as support vector machine will help physicians to diagnose accurately, I will explain the SVM algorithm step by step in this paper. In this study, Wisconsin diagnostic breast cancer data set is used to classify the tumors as Benign and Malignant. Performances of this classifier are evaluated by the model accuracy.

KEYWORDS: *Machine Learning, Deep Learning, Artificial Intelligence, Medicine, Breast Cancer, Cancer prognosis and prediction, Data Science, Support Vector Machine, Breast Cancer Classifier, Regression, Supervised Machine Learning*

I. INTRODUCTION

In this paper we will use 12 attributes, out of which 10 real valued features from each cell nucleus obtained from the Wisconsin diagnostic breast cancer dataset that explains about the stage of breast cancer M (Malignant) and B (Benign).

Features explanations:

1. ID: Patient id
2. Diagnosis (M = Malignant, B = Benign)
3. Radius(mean of distances from center to points on the perimeter) (worst). Worst texture. Texture (standard deviation of gray-scale values) (worst). Worst perimeter. perimeter (worst)
4. Texture (Breast cancer can cause changes and inflammation in skin cells that can lead to texture changes), here we can take the standard deviation of grey scaled values
5. Perimeter: Size of the core tumor
6. Area: Area of the core tumor
7. Smoothness: Local variation in radius length
8. Compactness: (perimeter^2 / area - 1.0)
9. Concavity (severity of concave portions of the contour)
10. Concave points (Number of concave portions of the contour)
11. Symmetry
12. Fractal Dimension ("coastline approximation" - 1)

Except the ID and Diagnosis, all other features are divided into three parts, the first part is Mean, that tells about the mean of all cells, the second part is Standard Error that tells about the standard error of the cells and the third part is worst mean of the worst cells. Now we have total of 30 features that will be the data for the Support Vector Machine Learning -Artificial Intelligence (Machine Learning & Deep Learning)

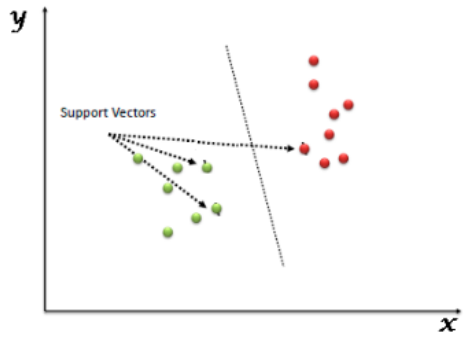
S.No	Features
1	radius_mean
2	texture_mean
3	perimeter_mean
4	area_mean
5	smoothness_mean
6	compactness_mean
7	concavity_mean
8	concave points_mean
9	symmetry_mean
10	fractal_dimension_mean
11	radius_se
12	texture_se
13	perimeter_se
14	area_se
15	smoothness_se
16	compactness_se
17	concavity_se
18	concave points_se
19	symmetry_se
20	fractal_dimension_se
21	radius_worst
22	texture_worst
23	perimeter_worst
24	area_worst
25	smoothness_worst
26	compactness_worst
27	concavity_worst
28	concave points_worst
29	symmetry_worst
30	fractal_dimension_worst

For example, field 1 is the Mean Radius, field 11 Standard Error Radius, 21 Worst mean Radius of the Tumor.

All feature values are recorded with four significant digits. Missing values are none (no missing values), class distribution 357 Benign, 212 Malignant.

II. Methodology

Support Vector Machine is a supervised machine learning algorithm, this machine learning technique is used to solve the both classification and regression problems. In this algorithm, we plot each data item as a point in n-dimensional space(where n is the number of features), here we have 30 features with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiate two classes



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/line)

Process of implementing SVM in python

Python has the inbuilt library Scikit-learn which is widely used library for implementing the machine learning algorithms, Support Vector machine is also available in python

We will implement the SVM algorithm using python and will explain every step

1. First step is to implement list of required libraries to build machine learning

```
from sklearn import metrics
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn import svm
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn import metrics
```

2. Importing the Wisconsin Breast Cancer Dataset into the python

```
dataset = pd.read_csv("C:\\Users\\skumarlocal\\Desktop\\AI(Machine Learning)\\Breast_cancer_Data_Analysis.csv")
dataset.head()
```

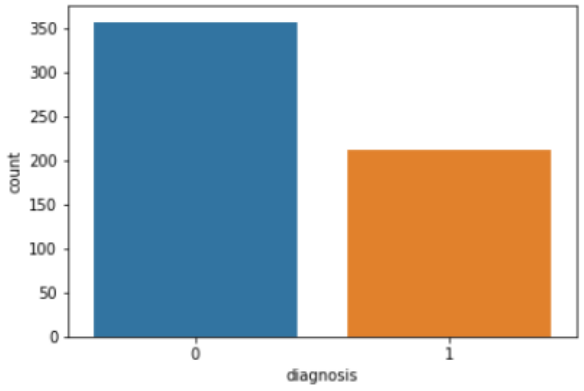
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	..
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	..
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	..

3. Based on our earlier explanation, the Wisconsin Breast Cancer Dataset divided into three parts (Mean, standard error, worst)

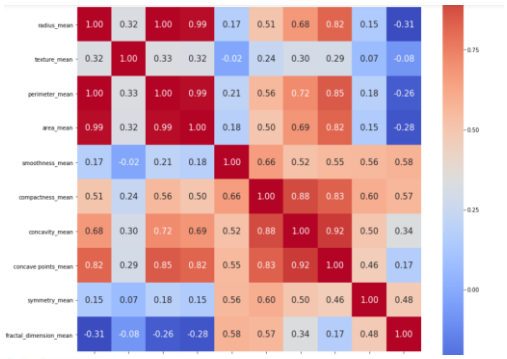
```
Explanatory_features_mean = list(dataset.columns[1:11])
Explanatory_features_se = list(dataset.columns[11:20])
Explanatory_features_worst = list(dataset.columns[21:31])
print(Explanatory_features_mean)
print(Explanatory_features_se)
print(Explanatory_features_worst)
```

```
['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean']
['fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se']
['fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst']
```

4. The Wisconsin Breast Cancer dataset has the Target Variable that tells every cell nuclei record is diagnosed with Malignant (M) or Benign (B) tumor, Let us convert the Target variable to Binary Boolean (1,0), here Malignant (M) is 1 and Benign (B) is 0 before we feed the data to SVM algorithm. Let us plot the graph to see which Class of cancer is maximum, from the below graph Benign cancer patient is higher than the Malignant cancer.



5. Now let us plot a heat map chart that shows the correlation graph that can remove multi collinearity it means the columns are depending on each other so we should avoid it because what is the use of using same column twice, let us check the correlation between the features



From the heat map, the observations are the radius, perimeter and area are highly correlated, so we can take any of the feature.
Compactness mean, Concavity mean, Concave point mean are highly correlated and hence compactness mean can be used. These features such as perimeter mean, texture mean, compactness mean, and symmetry mean are considered as good features for prediction.

6. Now let us test and train the dataset, so we can split 25% dataset as test data and remaining 75% dataset will be considered for training dataset.

```
train,test = train_test_split(dataset,test_size=0.25)
print(train.shape)
print(test.shape)
```

```
(426, 32)
(143, 32)
```

7. We will create a function that produces the accuracy of the model by using the Wisconsin data set for fitting the data

```
def classification_model(model,data,prediction_input,output):
    model.fit(data[prediction_input],data[output])
    predictions = model.predict(data[prediction_input])
    print("Accuracy : %s" % "{0:.3%}".format(accuracy))
    kf = KFold(data.shape[0], n_folds=5)
    error = []
    for train,test in kf:
        train_X = (data[prediction_input].iloc[train,:])
        train_y = data[output].iloc[train]
        model.fit(train_X, train_y)
        test_X=data[prediction_input].iloc[test,:]
        test_y=data[output].iloc[test]
        error.append(model.score(test_X,test_y))
        print("Cross-Validation Score : %s" % "{0:.3%}".format(np.mean(error)))
data_X= dataset[prediction_feature]
data_y= dataset["diagnosis"]
def Classification_model_gridsearchCV(model,param_grid,data_X,data_y):
    clf = GridSearchCV(model,param_grid,cv=10,scoring="accuracy")
    clf.fit(train_X,train_y)
    print("The best parameter found on development set is :")
    print(clf.best_params_)
    print("the bset estimator is ")
    print(clf.best_estimator_)
    print("The best score is ")
    print(clf.best_score_)
```

8. Now let us fit the data into the SVC Machine learning algorithm

```
model = svm.SVC()
param_grid = [
              {'C': [1, 10, 100, 1000],
               'kernel': ['linear']
              },
              {'C': [1, 10, 100, 1000],
               'gamma': [0.001, 0.0001],
               'kernel': ['rbf']
              },
 ]
Classification_model_gridsearchCV(model,param_grid,data_X,data_y)
```

9. The accuracy of the model predicts, that shows the 91% of the data classification is accurate.

```
The best parameter found on development set is :
{'C': 100, 'kernel': 'linear'}
the bset estimator is
SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
The best score is
0.9178403755868545
```

III. RESULTS

The Wisconsin Breast cancer data with 30 features was analyzed to identify the error rates and the accuracy of the model is 91.78%, which shows 91.78% is correct predictions.

IV. DISCUSSION

In this paper, we looked at the machine learning algorithm, Support Vector Machine in detail. I discussed its concept of working, process of implementation in python, the tricks to make the model efficient by tuning its parameters, I would suggest you to use SVM and analyze the power of this model by tuning the parameters. I also want to hear your experience with SVM, how have you tuned parameters to avoid over-fitting and reduce the training time?

REFERENCES

[1] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[2] https://www.kaggle.com/gargmanish/basic-machine-learning-with-cancer

[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[4] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data#data.csv

[5] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/version/1#README_data_info.rtf

[6] https://ieeexplore.ieee.org/document/6044334/references#references