Supervised Machine Learning Algorithms

Created by- Pavan Kumar M

Introduction

- Here we are trying to predict the probability of "*what percent of the diagnosis is Malignant or Benign from the chosen dataset using different machine learning algorithms like SVM, Naïve bayes and Neural Network*", and conclude which algorithm is better in real time application based on the type of dataset and the precision we get.

What is SVM?

- It is a classification type of supervised learning algorithms.
- Support Vector Machines(SVM) is a classification approach but can be employed in both classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM is less prone to outliers as it only cares about the points that are closest to decision boundary or support vector.
- The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes. In order to achieve MMH the margin between classes should have the same distance.

What is Naïve Bayes?

- It is a classification type of supervised learning algorithms.
- It is a classification technique based on Bayes’ Theorem, with an assumption of independence among the predictors.
- In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naïve Bayes is known to outperform even highly sophisticated classification methods.
- Naïve Bayes algorithm assess the probability of outcome based on the dependent events. It is easy to build and useful for large dataset.

What is Neural network?

- Neural Networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks.
- It has multiple input channels to accept training samples represented as a vector, and processing stage where the weights(w) are adjusted such that the output error (actual vs predicted) is minimized. Then the result is fed into an activation function to produce output.

Insights

- Recall- describes what percentage of positive cases did model catch.
  - Recall= TP/(FN+TP)
- Precision- describes what percentage of prediction was correct.
  - Precision=TP/(TP+FP)
- F1 score- it is the weighed average of precision and recall
  - F1 score= 2*((precision*recall)/(precision + recall))

True Negative(TN)-Actual FALSE that was predicted as FALSE

False Positive(FP)- Actual FALSE that was predicted as TRUE(Type I error)

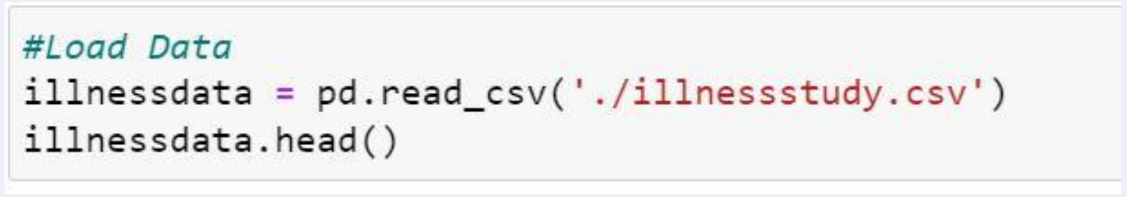
False Negative(FN)-Actual TRUE that was predicted as FALSE(Type II error)

True Positive(TP)-Actual TRUE that was predicted as TRUE

	PREDICTED		
ACTUAL	FALSE	TN	FP
	TRUE	FN	TP

Identifiers

- Here we are trying to find if the diagnosis is *Malignant or Benign* using the *'illnessdataset*" dataset.
- For that we need an identifier which denotes if the diagnosis is harmful or not. This is denoted by 'B' for Benign and 'M' for Malignant.

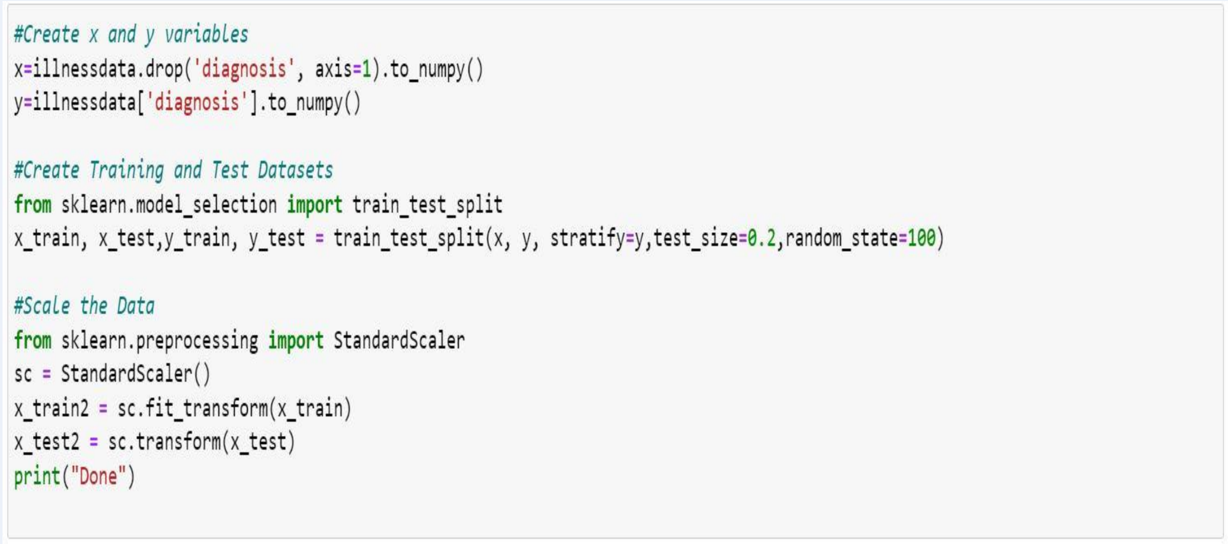


```
#Load Data
illnessdata = pd.read_csv('./illnessstudy.csv')
illnessdata.head()
```

Train and Test Model

- The independent variables is defined to x-axis and dependent variable *Diagnosis* is defined to y-axis.
- The dataset is divided into 2 types.
    - Train
    - Test

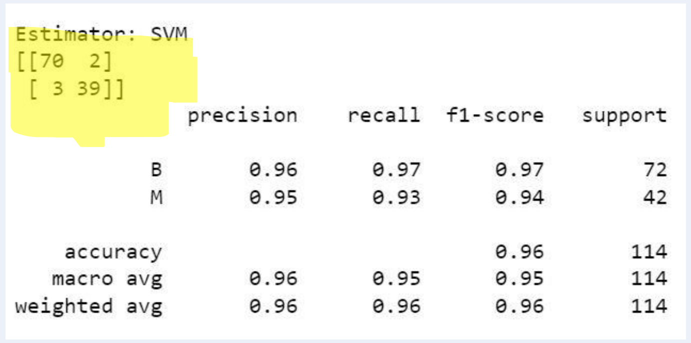
- The train dataset is used to build the model and test dataset is used to test the model.
- The test size is 0.2 which means the 20% of the dataset is used in training model and the rest is used to build/test the model.



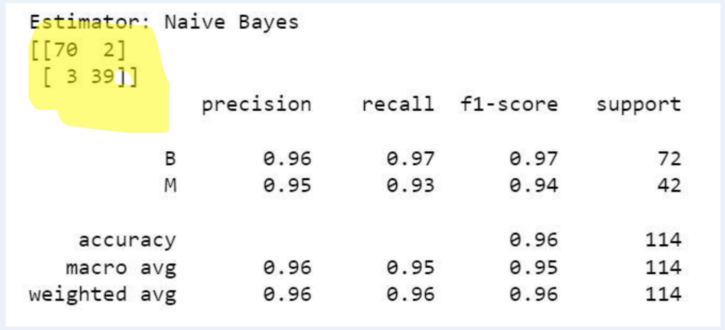
```
#Create x and y variables
x=illnessdata.drop('diagnosis', axis=1).to_numpy()
y=illnessdata['diagnosis'].to_numpy()

#Create Training and Test Datasets
from sklearn.model_selection import train_test_split
x_train, x_test,y_train, y_test = train_test_split(x, y, stratify=y,test_size=0.2,random_state=100)

#Scale the Data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train2 = sc.fit_transform(x_train)
x_test2 = sc.transform(x_test)
print("Done")
```

Confusion Matrices

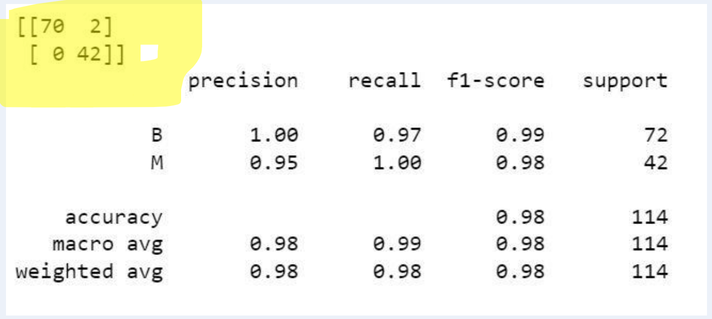
The confusion matrix is the table used for describing the performance of the model.



```
Estimator: SVM
[[70  2]
 [ 3 39]]
              precision    recall  f1-score   support

           B       0.96      0.97      0.97        72
           M       0.95      0.93      0.94        42

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```



```
Estimator: Naive Bayes
[[70  2]
 [ 3 39]]
              precision    recall  f1-score   support

           B       0.96      0.97      0.97        72
           M       0.95      0.93      0.94        42

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```



```
[[70  2]
 [ 0 42]]
              precision    recall  f1-score   support

           B       1.00      0.97      0.99        72
           M       0.95      1.00      0.98        42

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114
```

		PREDICTED	
ACTUAL	FALSE	TN=70	FP=2
	TRUE	FN=3	TP=39

SVM

		PREDICTED	
ACTUAL	FALSE	TN=70	FP=2
	TRUE	FN=3	TP=39

Naïve bayes

		PREDICTED	
ACTUAL	FALSE	TN=70	FP=2
	TRUE	FN=0	TP=42

Neural network

SVM

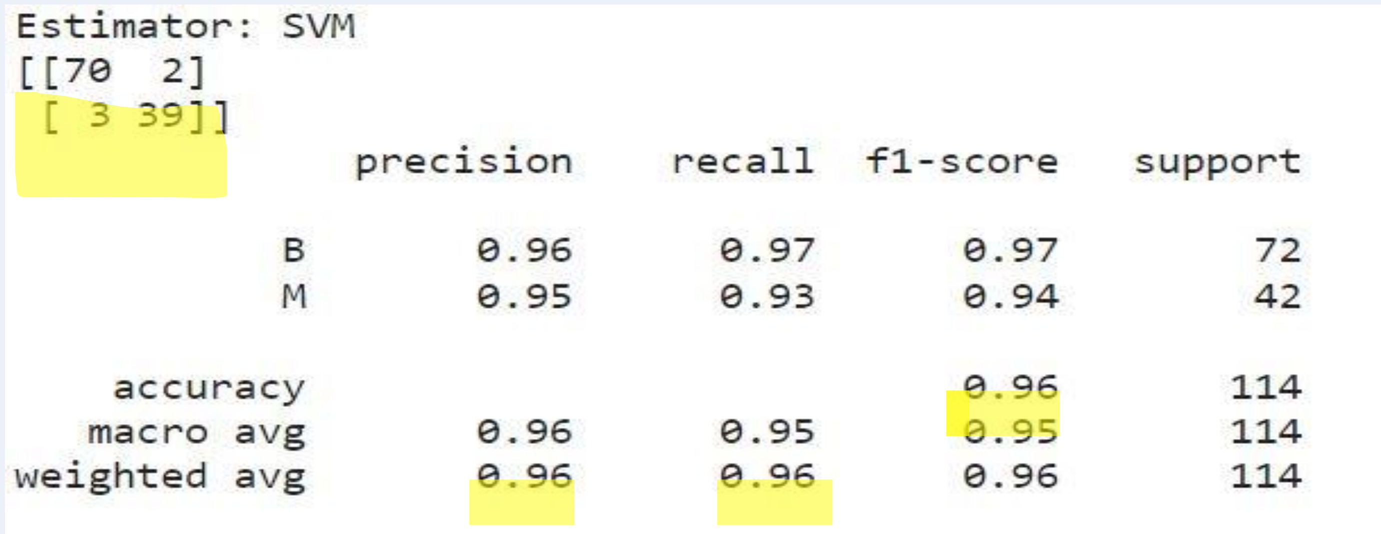
- Size of the sample- ‘B’=72, and ‘M’=42
  - Here, for B, the model classified 70 correctly and 2 incorrectly.
  - And, for M, the model classified 3 incorrectly and 39 correctly.

The **precision** of the SVM is 96% i.e., the model is 96% accurate out of positive prediction.

Recall=95% i.e., our model captured 95% of the actual positives considering them as true positives

F1 score is 96% i.e., our model is 96% accurate or in other words it gave an error of 1-0.96= 4%

This implies that our SVM model predicted ***Benign*** **of Diagnosis as 96%.**



```
Estimator: SVM
[[70  2]
 [ 3 39]]
              precision    recall  f1-score   support

           B       0.96      0.97      0.97        72
           M       0.95      0.93      0.94        42

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```

Naïve Bayes

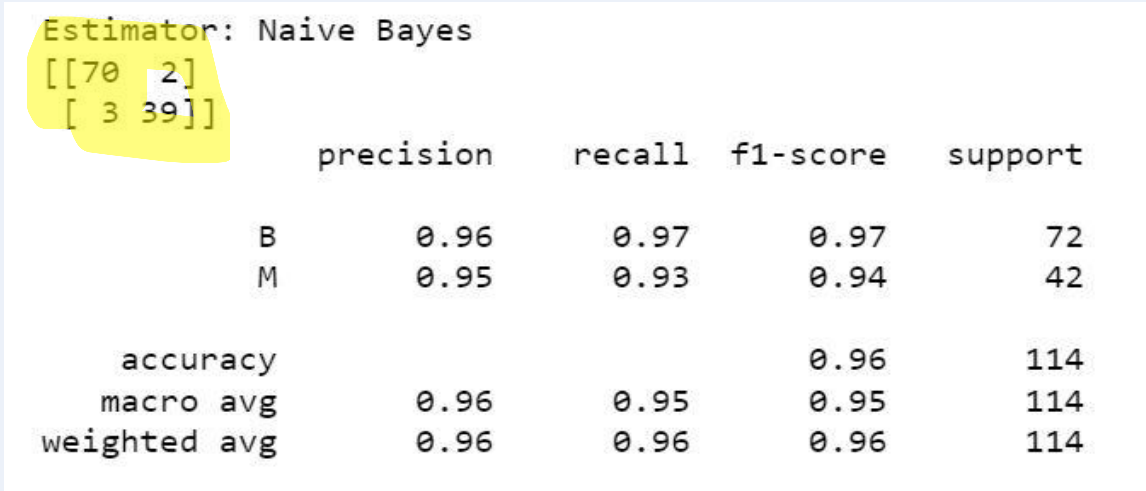
- Size of the sample- ‘B’=72, and ‘M’=42
  - Here, for B, the model classified *70 correctly and 2 incorrectly*.
  - And, for M, the model classified *3 incorrectly and 39 correctly*.

The **precision** of the Naïve Bayes is 96% i.e., the model is 96% accurate out of positive prediction.

Recall=95% i.e., our model captured 95% of the actual positives considering them as true positives

F1 score is 96% i.e., our model is 96% accurate or in other words it gave an error of 1-0.96= 4%

This implies that our Logistic regression model predicted ***Benign* of Diagnosis as 96%.**



```
Estimator: Naive Bayes
[[70  2]
 [ 3 39]]
              precision    recall  f1-score   support

           B       0.96      0.97      0.97        72
           M       0.95      0.93      0.94        42

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```

Neural network

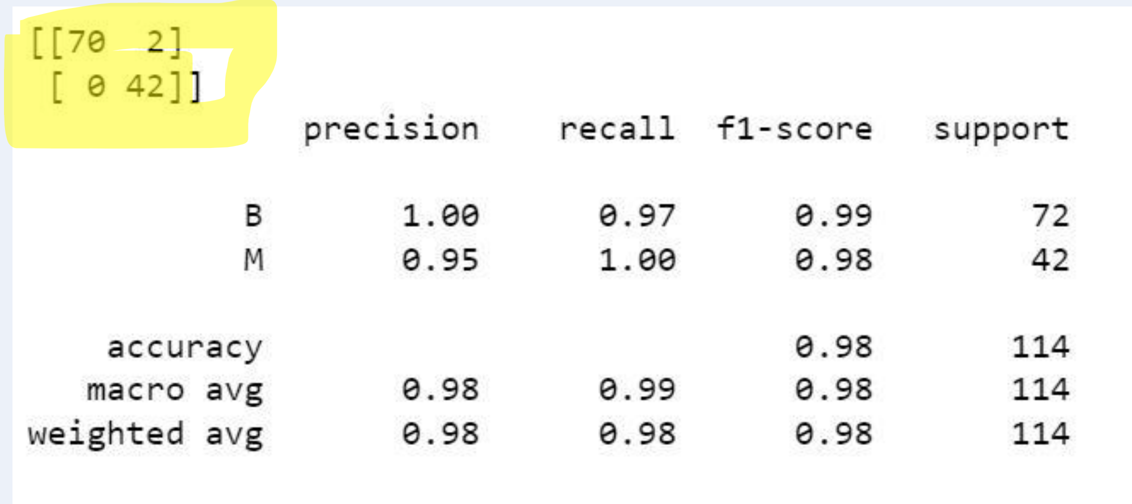
- Size of the sample- ‘B’=72, and ‘M’=42
  - Here, for B, the model classified 70 correctly and 2 incorrectly.
  - And, for M, the model classified 0 incorrectly and 42 correctly.

The **precision** of the neural network is 98% i.e., the model is 98% accurate out of positive prediction.

Recall=98% i.e., our model captured 98% of the actual positives considering them as true positives

F1 score is 98% i.e., our model is 98% accurate or in other words it gave an error of 1-0.98= 2%

This implies that our <u>neural network</u> <u>model predicted</u> <u>*Benign* of Diagnosis as 98%.</u>



```
[[70  2]
 [ 0 42]]
              precision    recall  f1-score   support

           B       1.00      0.97      0.99        72
           M       0.95      1.00      0.98        42

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114
```

Recommendation

- We can notice from the results that the accuracy of *Neural network* is higher than *SVM and Naïve bayes*.
- For the same number of samples chosen, SVM and Naïve Bayes algorithms was able to be 96% precise where as, Neural Network algorithm was 98% precise. i.e., the results of Neural network is 2% better than SVM and Naïve Bayes.
- As we require only classification results *(Benign or Malignant)* and Neural network can do the same with simple algorithm*(Simple the algorithm, lesser will be the time taken to produce the results with large datasets)*, **Neural Network is recommended** as the most suitable algorithm to predict the diagnosis effect.

Scope for model Improvement

More data

Having more datapoints helps in achieving better precision. More datapoints increases more datapoints for training model and this helps in achieving better precision.

Increasing the sample size

The increase in sample size improves the accuracy of the results for relatively large dataset and analysis which involves logistic regression.

Increased number of iterations

Higher number of iterations converges the model towards more precision and accuracy.
For our model we chose the number of iteration to be 10,000. And if we choose higher iterations, it converges our model towards more accuracy

Never forget these basic statistics !!

- **Mean** - The mean is the sum of all data points divided by total number of data points.
  - The mean is the way to measure the center of a distribution of data, i.e., it gives the central value of the dataset.
- **Median** or **50th Quartile-** it is the middle value of all the data points. It can also be referred as Median.
  - It divides the data into half.
  - It gives us an idea where the central value of the dataset is located.

- **Standard Deviation-** It is the square root of the variance. Standard deviation measures the offset of data relative to its mean.
  - **Variance-** It measures how far the observation lies away from the mean.
  - ***Standard deviation is useful to understand how far away the observation is from the mean.***

- **25th & 75th percentile**
  - 25% of the data falls below this 25th percentile.
  - 75% of the data falls below this 75th percentile.
  - **The percentiles helps in knowing where a particular data point lies within the data set**.
- **Min & Max-** It is the minimum and maximum value of the any given parameter in a dataset.
  - This helps in knowing the range of values for the given parameter/variable in a dataset.
  - **MIN & MAX values helps to understand if other statistical values fall with this range.**