International Journal of Engineering & Technology, 7 (2.8) (2018) 315-319

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

A Comparative study of machine learning algorithms on thyroid disease prediction

[1]Shaik Razia, [2]P. Swathi Prathyusha, [3]N. Vamsi Krishna, [4]N. Sathya Sumana

[1]Associate Professor, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India 522502
[2, 3, 4] UG Students, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India 522502
**Email: razia28sk@gmail.com*

Abstract

Thyroid illness is a medicinal state that influences the functionality of the thyroid organ that is thyroid gland [1](Guyton, 2011).The indications of thyroid ailment differ basing upon the type. There are four most common varieties: hypothyroidism (low capacity) which is caused due to the insufficiency of the thyroid hormones; hyperthyroidism (high capacity) which is caused due to the existence of the thyroid hormones more than just sufficient, basic variations from the norm, most normally an augmentation of the thyroid organ; and tumors which can be benign or can cause cancer. It is additionally conceivable to have irregular thyroid capacity tests with no clinical side effects [2](Bauer & al, 2013). In this study a comparative thyroid disease diagnosis were performed by using Machine learning techniques that is Support Vector Machine (SVM), Multiple Linear Regression, Naïve Bayes, Decision Trees. For this purpose, thyroid disease dataset gathered from the UCI machine learning database was used.

Keywords: *SVM, Multiple Linear Regression, Naïve Bayes, Decision Trees.*

1. Introduction

Thyroid issues are impact on the thyroid organ, it is a butterfly shaped organ within the front of the neck.

The thyroid has basic components to coordinate totally different metabolic ways during the body. Different types of thyroid issue impact either its structure or limit.

The thyroid organ is organized beneath the Adam's apple wrapped around the trachea (windpipe). A small tissue of inside the organ called as the isthmus; attach the two thyroid projections on all sides. The thyroid utilizes iodine to convey the major hormones. Thyroxine, usually known as T4, is the essential hormone made by the organ. When transport by ways for the dissemination framework to the body's tissues, a small piece of the T4 released from the organ is modified over to triiodothyronine (T3) that is the most unique hormone.

The cerebrum includes the limit of the thyroid organ is overseen by an info framework. When thyroid hormone levels are precisely low, the hypothalamus in the cerebrum transfer a hormone known as thyrotropin releasing hormone(TRH) that causes the pituitary organ (arranged at the base of the brain) to release thyroid invigorating hormone (TSH). TSH enables the thyroid organ to release more T4.

1.1 Dataset Description: [3]

Attribute	**Data Type**	**Value Range**
Age	Real	[0.00,0.93]
Sex	Integer	[0,1]
On_thyroxine	Integer	[0,1]
Query_on_thyroxine	Integer	[0,1]
antithyroid_medication	Integer	[0,1]
Sick	Integer	[0,1]
Pregnant	Integer	[0,1]
Thyroid_surgery	Integer	[0,1]
I131_treatment	Integer	[0,1]
Query_hypothyroid	Integer	[0,1]
Query_hyperthyroid	Integer	[0,1]
Lithium	Integer	[0,1]
Goitre	Integer	[0,1]
Tumor	Integer	[0,1]
Hypopituitary	Integer	[0,1]
Psych	Integer	[0,1]
TSH	Real	[0.0, 0.53]
T3	Real	[.0005,.18]
TT4	Real	[0.0020, 0.6]
T4U	Real	[0.017, 0.233]
FTI	Real	[0.0020, 0.642]
Class	Integer	{1,2,3}

Where class is varied as following
Normal - 1
Hyperthyroidism - 2
Hypothyroidism - 3

1.2 Implementation

As the data set has no missing values at the pre-processing stage the data set is imported using the pandas library in python, and it is then splitted into training set and test set which consists of 5760 and 1440 observations respectively and they are tested on various algorithms of machine learning and the split as of randomness and the predicted values are compared and the accuracy of the model or the algorithm is calculated using the below formula

$$Accuracy = \frac{TP + TN}{n} * 100$$

Where TP=True Positives
TN=True Negatives
n=Number of observations in test set

To find some of true positives and true negatives we have to find the sum of principal diagonal in confusion matrix. The confusion matrix is a matrix of n*n rows and columns that depend upon the elements in output class. In this case the confusion matrix is of 3*3 matrix.

2. Multiple Linear Regressions

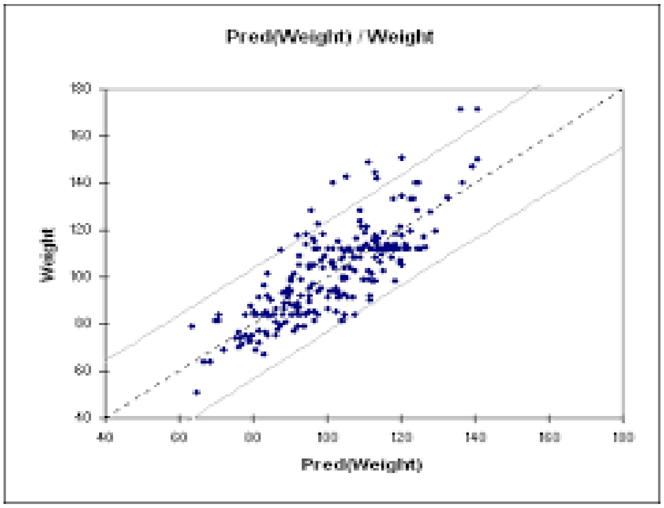
It is augmentation of simple linear regression. It takes the attributes as input data sources and delivers a yield.

MLR examines the connection between at least two IVs and a solitary DV, where IV is an independent variable and DV is a dependent variable.

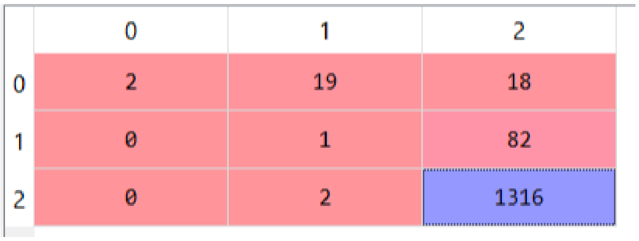
Y=x1+x2+...+xi

Where y is a dependent variable which depends upon x1,x2,x3……,xi which are independent variables.

TP+TN=1319



2.1 Confusion matrix of Multiple Linear Regressions

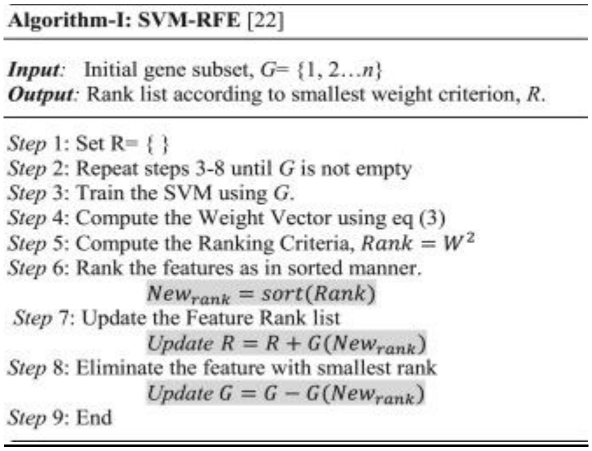


	0	1	2
0	2	19	18
1	0	1	82
2	0	2	1316

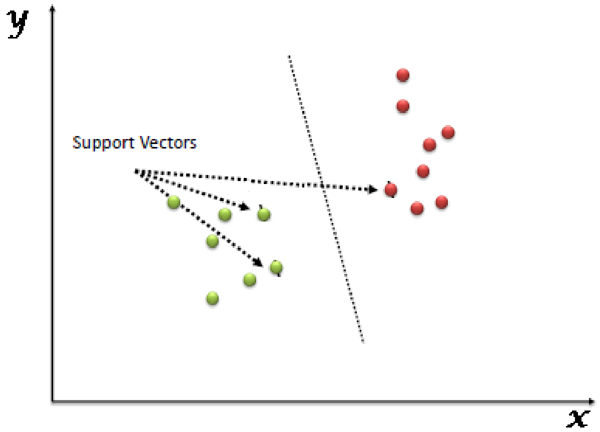
By using multi linear regression we can get the correctness of **91.59%**

3. Svm (Support Vector Machine)

Support Vector Machine (SVM) is a managed machine learning count which can be used for both classification and regression issues. Regardless, it's usually used as a bit of arrangement problems. During this figuring, we have a tendency to plot every data point factor as some extent in n-dimensional area (where n is number of qualities you have) with the estimation of each half being the estimation of a selected organize. By then, we have a tendency to perform characterization by finding the hype-plane that completely different the two categories notably well.

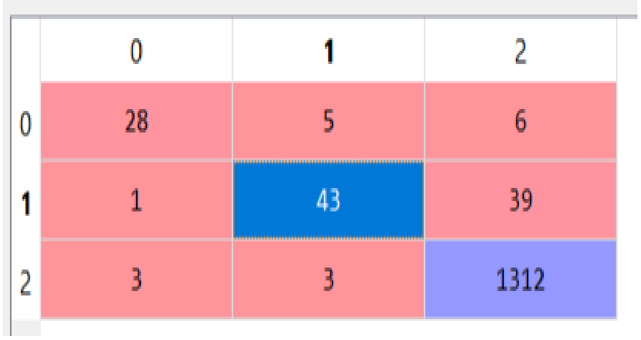


```
Algorithm-I: SVM-RFE [22]
Input: Initial gene subset, G= {1, 2…n}
Output: Rank list according to smallest weight criterion, R.
Step 1: Set R= { }
Step 2: Repeat steps 3-8 until G is not empty
Step 3: Train the SVM using G.
Step 4: Compute the Weight Vector using eq (3)
Step 5: Compute the Ranking Criteria, $Rank = W^2$
Step 6: Rank the features as in sorted manner.
        $New_{rank} = sort(Rank)$
Step 7: Update the Feature Rank list
        $Update\ R = R + G(New_{rank})$
Step 8: Eliminate the feature with smallest rank
        $Update\ G = G - G(New_{rank})$
Step 9: End
```



TP+TN=1383

2.1 Confusion matrix of SVM



	0	1	2
0	28	5	6
1	1	43	39
2	3	3	1312

By using Support vector machine we can get the accuracy of **96.04%**

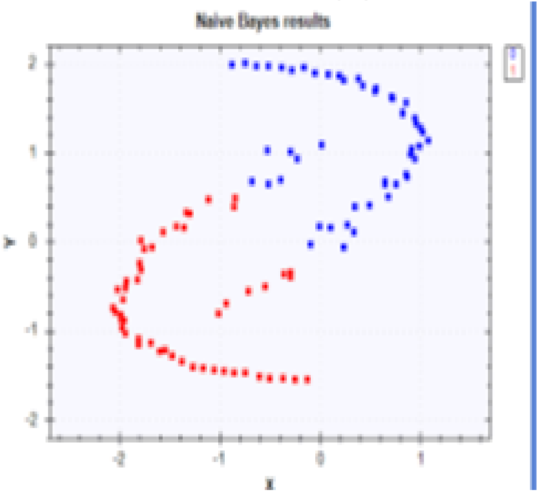
4. Naive Bayes

Naive Bayes classifiers are an accumulation of classification methods based on Bayes' Theorem. It isn't a solitary algorithms yet a group of algorithms where every one of them share a typical guideline, i.e. each combination of attributes being classified is autonomous of each other.
The dataset is divided into two different categories like **feature matrix** and **response vector.**
→The feature matrix consists each and every vector that is nothing but row of dataset in which every vector comprises of the estimate of dependent features.
→Response vector contains the esteem of class variable (prediction or yield) for each vector which is nothing but a row of the feature matrix.

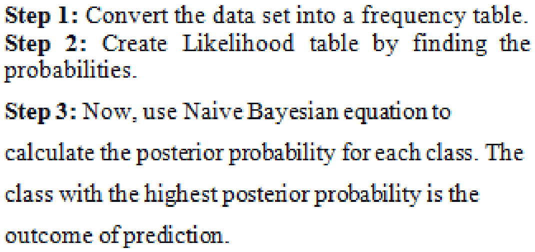
4.1 Bayes Theorem

Bayes' Theorem finds the likelihood of an event happening given the likelihood of another event that has just happened. Bayes' hypothesis is expressed numerically as the following condition:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$



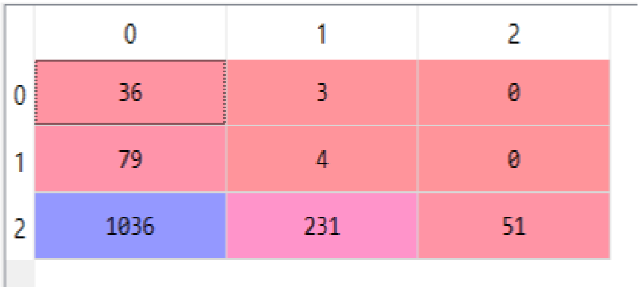
Algorithm


Step 1: Convert the data set into a frequency table.
Step 2: Create Likelihood table by finding the probabilities.

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

TP+TN=91

4.2 Confusion matrix of Naïve Bayes

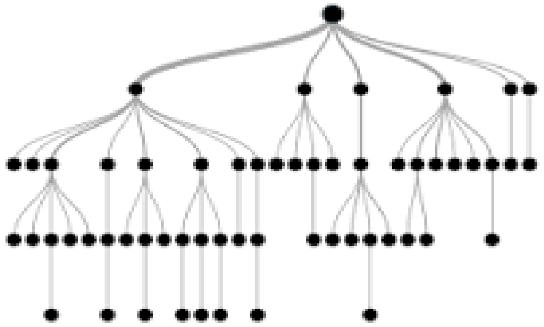


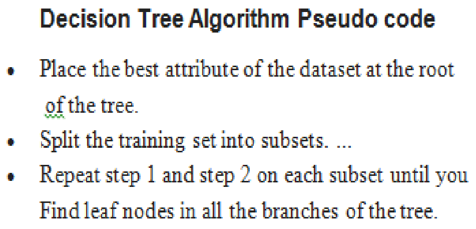
	0	1	2
0	36	3	0
1	79	4	0
2	1036	231	51

By using Naïve Bayes algorithm we can get the exactness of **6.31%.** So Naïve Bayes has the least accuracy.

5. Decision Trees

The Decision tree is one of the classification methods. This learning calculation applies a separation and conquer methodology that can also be called as divide and conquer to build the tree. The arrangements of instances are related by an arrangement of attributes.
A Decision tree includes hubs and leaves, where hubs represent a test on the estimations of a trait and leaves represent the class of an instance that fulfills the conditions.
The result is "true" or "false" that is nothing but a categorical variable. Standards can be gotten from the way beginning from the root hub to the leaf and using the hubs on way as preconditions for the rule, to foresee the class at the leaf. The tree pruning must be done to expel pointless preconditions and duplications.

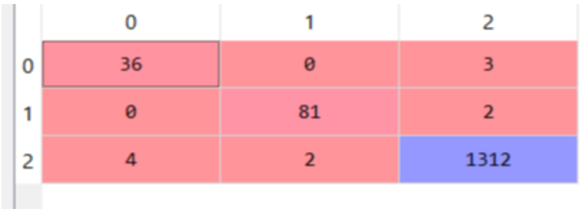



Decision Tree Algorithm Pseudo code

- Place the best attribute of the dataset at the root of the tree.
- Split the training set into subsets. ...
- Repeat step 1 and step 2 on each subset until you Find leaf nodes in all the branches of the tree.

TP+TN=1429

Confusion matrix of Decision trees



	0	1	2
0	36	0	3
1	0	81	2
2	4	2	1312

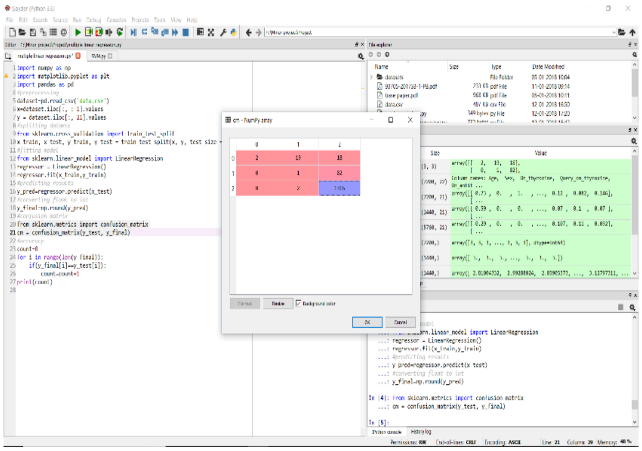
By using Decision tree algorithm we can get the highest correctness of **99.23%.**

5. Results

5.1 The comparison of Results

Algorithm	Accuracy
Multiple Linear Regression	91.59%
SVM	96.04%
Naïve Bayes	6.31%
Decision Trees	99.23%

5.2 Multi Linear Regression output



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#preprocessing
dataset=pd.read_csv('data.csv')
x=dataset.iloc[:, :-1].values
y = dataset.iloc[:, 21].values
#splitting dataset
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
#fitting model
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
#predicting results
y_pred=regressor.predict(x_test)
#converting float to int
y_final=np.round(y_pred)
#confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_final)
#accuracy
count=0
for i in range(len(y_final)):
    if(y_final[i]==y_test[i]):
        count=count+1
print(count)
```

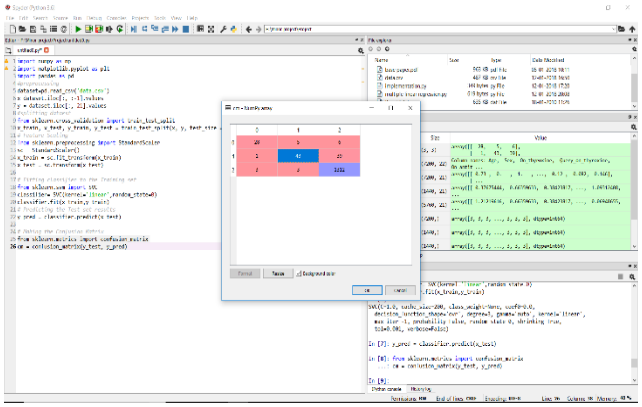
cm - NumPy array

	0	1	2
0	2	19	18
1	0	1	82
2	0	2	1316

```
In [4]: from sklearn.metrics import confusion_matrix
   ...: cm = confusion_matrix(y_test, y_final)

In [5]:
```

5.3 Support Vector Machine output



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#preprocessing
dataset=pd.read_csv('data.csv')
x=dataset.iloc[:, :-1].values
y = dataset.iloc[:, 21].values
#splitting dataset
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

# Fitting classifier to the Training set
from sklearn.svm import SVC
classifier= SVC(kernel='linear',random_state=0)
classifier.fit(x_train,y_train)
# Predicting the Test set results
y_pred = classifier.predict(x_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

cm - NumPy array

	0	1	2
0	28	5	6
1	1	41	39
2	4	5	1311

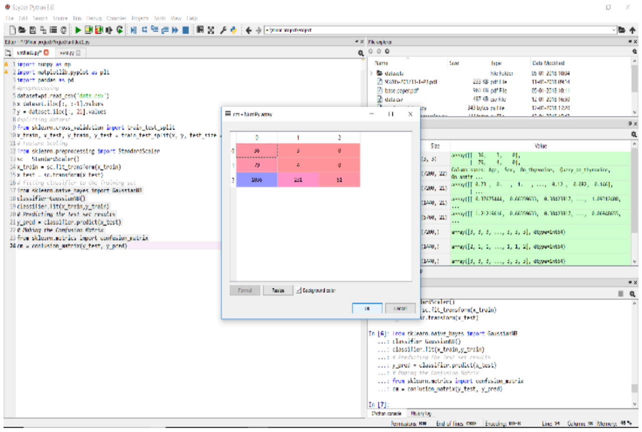
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=0, shrinking=True,
  tol=0.001, verbose=False)

In [7]: y_pred = classifier.predict(x_test)

In [8]: from sklearn.metrics import confusion_matrix
   ...: cm = confusion_matrix(y_test, y_pred)

In [9]:
```

5.4 Naïve Bayes output



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#preprocessing
dataset=pd.read_csv('data.csv')
x=dataset.iloc[:, :-1].values
y = dataset.iloc[:, 21].values
#splitting dataset
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
# Fitting classifier to the Training set
from sklearn.naive_bayes import GaussianNB
classifier=GaussianNB()
classifier.fit(x_train,y_train)
# Predicting the Test set results
y_pred = classifier.predict(x_test)
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

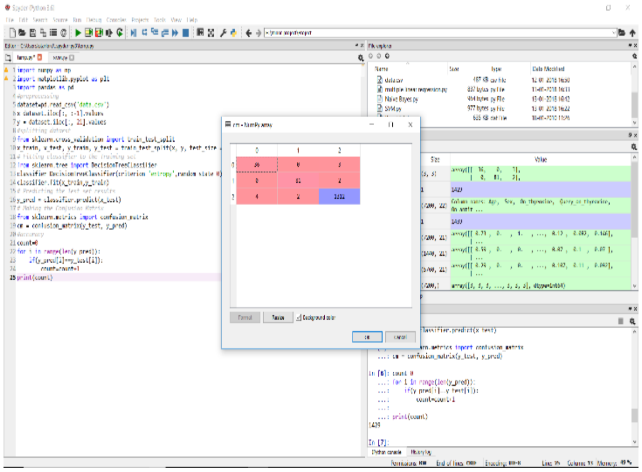
cm - NumPy array

	0	1	2
0	36	3	0
1	79	4	0
2	1036	251	51

```
In [6]: from sklearn.naive_bayes import GaussianNB
   ...: classifier=GaussianNB()
   ...: classifier.fit(x_train,y_train)
   ...: # Predicting the Test set results
   ...: y_pred = classifier.predict(x_test)
   ...: # Making the Confusion Matrix
   ...: from sklearn.metrics import confusion_matrix
   ...: cm = confusion_matrix(y_test, y_pred)

In [7]:
```

5.5 Decision Tree output



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#preprocessing
dataset=pd.read_csv('data.csv')
x=dataset.iloc[:, :-1].values
y = dataset.iloc[:, 21].values
#splitting dataset
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
# Fitting classifier to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(criterion='entropy',random_state=0)
classifier.fit(x_train,y_train)
# Predicting the Test set results
y_pred = classifier.predict(x_test)
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
#accuracy
count=0
for i in range(len(y_pred)):
    if(y_pred[i]==y_test[i]):
        count=count+1
print(count)
```

cm - NumPy array

	0	1	2
0	36	0	3
1	0	81	2
2	4	2	1312

```
   ...: cm = confusion_matrix(y_test, y_pred)

In [6]: count=0
   ...: for i in range(len(y_pred)):
   ...:     if(y_pred[i]==y_test[i]):
   ...:         count=count+1
   ...:
   ...: print(count)
1429

In [7]:
```

6. Conclusion

This paper presents a comparative study on thyroid disease diagnosis by using Support Vector Machine (SVM), Multiple Linear Regression, Naïve Bayes and Decision Trees. The results were compared and it was seen that Decision Trees could be successfully used to help the diagnosis of thyroid disease. It is observed that the Decision Trees outperformed the SVM, Multi linear regression, Naïve Bayes with respect to the accuracy of the network to diagnose the thyroid disease.

References

[1] C., Guyton, Arthur. Guyton and Hall textbook of medical physiology, OCLC 434319356

[2] Bauer, DC; et al. (2O13). Path physiology of Disease: An Introduction to Clinical Medicine, Seventh Edition, New York, NY: McGraw-Hill – via Access Medicine.

[3] http://sci2s.ugr.es/keel/dataset.php?cod=67

[4] Keleş A, Keleş A. Estdd: Expert system for thyroid dis-eases diagnosis. Expert Systems with Applications. 2OO8; 34:242–6.

[5] Ozyilmaz L, Yildirim T. Diagnosis of thyroid diseaseusing artificial neural network methods. Proceedings of the 9th International Conference on Neural Information Processing, 2OO2. ICONIP’O2; 2OO2. p. 2O33–6.

[6] Temurtas F. A comparative study on thyroid disease diagnosis using neural networks. Expert Systems with Applications. 2OO9; 36:944–9.

[7] Hasan Makas, Nejat Yumusak (2ol3) A Comprehensive Study On Thyroid Diagnosis By Neural Networks And Swarm Intelligence. International Conference On Electronics, Computer And Computation (Icecco):L8o-L84.Doi:Lo.Llo9/Icecco.2ol3.6718258

[8] V. Prasad, T. Srinivasa Rao, M. Surendra Prasad Babu, Thyroid Disease Diagnosis Via Hybrid Architecture Composing Rough Data Sets Theory And Machine Learning Algorithms. Soft Computing 2o(3):Ll79-Ll89 2ol5.

[9] Li-Na Li, Ji-Hong Ouyang, Hui-Ling Chen, Da-You Liu, A Computer Aided Diagnosis System For Thyroid Disease Using Extreme Learning Machine. Journal Of Medical Systems 36(5):3327-3337 2ol2

[10] Shaik.Razia, M.R.Narasingarao Published “A Neuro Computing Frame Work For Thyroid Disease Diagnosis Using Machine Learning Techniques”, Vol.95. No.9. Pages L996-2oo5, Issn: L992-8645.

[11] P. Gopi Krishna, K. Sreenivasa Ravi “Designing A Multipurpose Reconfigurable Wireless Node For Broadcasting And Unicasting In Rereal Time Applications” In International Journal Of Pure And Applied Mathematics (Ijpam). Volume Ll5 No. 8 2ol7, 5o5-5lo.

[12] P Gopi Krishna, K Sreenivasa Ravi “Implementation Of Mqtt Protocol On Low Resourced Embedded Network” In International

Journal Of Pure And Applied Mathematics (Ijpam). Volume Ll6 No. 6 2ol7, L6l-L66.

[13] Dr. Seetaiah Kilaru, Hari Kishore K, Sravani T, Anvesh Chowdary L, Balaji T “Review And Analysis Of Promising Technologies With Respect To Fifth Generation Networks”, 2ol4 First International Conference On Networks & Soft Computing, Issn:978-L-4799-34867/L4,Pp.27o-273,August2ol4.

[14] N.Prathima, K.Hari Kishore, “Design Of A Low Power And High Performance Digital Multiplier Using A Novel 8t Adder”, International Journal Of Engineering Research And Applications, Issn: 2248-9622, Vol. 3, Issue.L, Jan-Feb., 2ol3.

[15] T. Padmapriya And V. Saminadan, “Improving Throughput For Downlink Multi User Mimo-Lte Advanced Networks Using Sinr Approximation And Hierarchical Csi Feedback”, International Journal Of Mobile Design Network And Innovation- Inderscience Publisher, Issn : L744-285o Vol. 6, No.L, Pp. L4-23, May 2ol5.

[16] S.V.Manikanthan And K. Srividhya “An Android Based Secure Access Control Using Arm And Cloud Computing", Published In: Electronics And Communication Systems (Icecs), 2ol5 2nd International Conference On 26-27 Feb. 2015,Publisher: Ieee, Doi: Lo.Llo9/Ecs.2ol5.7l24833.

[17] M. Rajesh, Manikanthan, “Annoyed Realm Outlook Taxonomy Using Twin Transfer Learning”, International Journal Of Pure And Applied Mathematics, Issn No:L3l4-3395, Vol-Ll6, No. 2l, Oct 2ol7.

[18] [18] K.Srikar, M.Akhil, V.Krishna Reddy “Execution Of Cloud Scheduling Algorithms” International Innovative Research Journal Of Engineering And Technology Issn No: 2456-L983.Volume 2, Issue 4 June 2ol7.

[19] Meherwar Fatima, M. P. (2ol7). Survey Of Machine Learning Algorithms For Disease Diagnostic. Journal Of Intelligent Learning Systems And Applications, L-L6.

[20] Jian, A. (2ol5). Machine Learning Techniques For Medical Diagnosis. Icstat.

[21] Alic, B. (2ol7). Machine Learning Techniques For Classification Of Diabetes And Cardiovascular Diseases. Mediterranean Conference On Embedded Computing.

[22] Shaik Razia, M.R.Narasingarao, Polaiah Bojja Published “Development And Analysis Of Support Vector Machine Techniques For Early Prediction Of Breast Cancer And Thyroid” In Scopus Indexed Journal Jardcs (Journal Of Advanced Research In Dynamical And Control Systems, 2017.Vol.9.Sp.Issue:6 Issn: 1943-023x Page No: 869-878).

[23] Shaik Razia Published “A Review On Disease Diagnosis Using Machine Learning Techniques” International Journal Of Pure And Applied Mathematics, 2017, Volume 117, No. 16 2017, 79-85, Issn: 1311-8080.

[24] Shaik Razia, M.R.Narasingarao, Polaiah Bojja Published “The Analysis Of Data Representation Techniques For Early Prediction Of Breast Cancer” International Journal Of Pure And Applied Mathematics, 2017, Volume-115, Issue: 6, Issn: 1311-8080, Issn: 1314-3395 Page No: 177-183.

[25] Shaik Razia, M.R.Narasingarao Published “Machine Learning Techniques For Thyroid Disease Diagnosis - A Review” Indian Journal Of Science And Technology, Issn: 09746846, Volume-9, Issue 28, July 2016.

[26] Shaik Razia, M.R.Narasingarao,G R Sridhar Published “A Decision Support System For Prediction Of Thyroid Disease- A Comparison Of Multilayer Perception Neural Network And Radial Basis Function Neural Network” Journal Of Theoretical And Applied Information Technology, 31st October 2015. Vol.80. No.3, Issn: 1992-8645.

[27] Meka Bharadwaj, Hari Kishore "Enhanced Launch-Off-Capture Testing Using Bist Designs” Journal Of Engineering And Applied Sciences, Issn No: 1816-949x, Vol No.12, Issue No.3, Page: 636-643, April 2017.

[28] P Bala Gopal, K Hari Kishore, R.R Kalyan Venkatesh, P Harinath Mandalapu “An Fpga Implementation Of On Chip Uart Testing With Bist Techniques”, International Journal Of Applied Engineering Research, Issn 0973-4562, Volume 10, Number 14 , Pp. 34047-34051, August 2015.