Comparing Decision Tree, Support Vector Machine and Neural Networks for Solving Classification Problem brought by Occupancy Detection

Jigyasa Kohli (jk423), Mahesh Muraleedharan Nair (mmn9), Venkata Jayasimha Hari Chidiri (vc279)
Group 3, MGMT 635 851
Information Systems Department,
New Jersey Institute of Technology, Newark.

Abstract

In this paper, an approach to solving classification problems by using decision trees, Support Vector Machine , neural networks system is elaborated. Field of Information theory is used to select a set of important attributes that can be used to classify tuples. Data mining topics will be discussed and a well dataset of Occupancy Detection Dataset will be used to create a neural network and churn this system's performance.

TECHNIQUES AND APPROACHES

1. Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.The deeper the tree, the more complex the decision rules and the fitter the model.

"In general, decision trees represent a disjunction of conjunctions of constraints on the attribute-values of instances. Each path from the tree root to a leaf corresponds to a conjunction of attribute tests, and the tree itself to a disjunction of these conjunctions" (Mitchell, 1997, p.53). Until each leaf node is populated by as homogeneous a sample set as possible:

General Form

Select a leaf node with an inhomogeneous sample set. Replace that leaf node by a test node that divides the inhomogeneous sample set into minimally inhomogeneous subsets, according to an entropy calculation.

Specific Form

Examine the attributes to add at the next level of the tree using an entropy calculation. Choose the attribute that minimizes the entropy.

Implementation Of Decision Tree in Python

In Python, scikit-learn is a widely used library for implementing machine learning algorithms, Decision Tree is also available in scikit-learn library and follow the same structure (Import

library, object creation, fitting model and prediction). Let's look at the below code this has been performed on the Occupancy Detection chosen for this project :

```
#Import Library

import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.cross_validation import train_test_split

#Assumed you have, X (predictor) and Y (target) for training data set and
X_test(predictor) of test_dataset
# Create Decision Tree classification object

decisionTree = DecisionTreeClassifier(max_depth=3)
#Train the model using the training sets and check score
decisionTree.fit(X_train, y_train)
#Predicting the score
predictions=decisionTree.predict(X_test)
print ("Accuracy of Decision Tree",accuracy_score(y_test,predictions))
```

Some advantages of decision trees are:

Simple to understand and to interpret. Trees can be visualised. Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. Able to handle both numerical and categorical data.

Other techniques are usually specialised in analysing datasets that have only one type of variable. Able to handle multi-output problems. Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

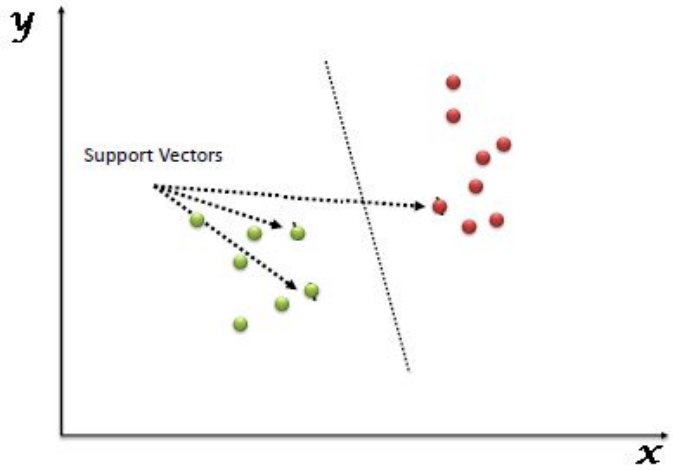
Some disadvantages of decision trees are:

Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting. Mechanisms such as pruning (not currently supported), setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble. The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree.

Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

2. Support Vector Machine

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems without any assumptions on the data distribution. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiate the two classes very well (look at the below snapshot).



Support Vectors are simply the coordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyperplane/ line).

Working of Support Vector Machine

We got a brief idea about the process of segregating the two classes with a hyper-plane. Let's understand How can we identify the right hyperplane?.

A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Implementation Of Support Vector Machine in Python

In Python, scikit-learn is a widely used library for implementing machine learning algorithms, SVM is also available in scikit-learn library and follow the same structure (Import library, object creation, fitting model and prediction). Let’s look at the below code this has been performed on the Occupancy Detection chosen for this project :

```
#Import Library
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cross_validation import train_test_split
from sklearn.svm import SVC
#Assumed you have, X (predictor) and Y (target) for training data set and
X_test(predictor) of test_dataset
# Create SVM classification object
clfs=SVC(kernel='rbf',gamma=2, C=1)
# there is various option associated with it, like changing kernel, gamma and
C value. Will discuss more # about it in next section.Train the model using
the training sets and check score

clfs.fit(X_train, y_train)
#Prediction Of Score
y_pred=clfs.predict(X_test)
zxc=accuracy_score(y_test,y_pred)
print('Accuracy of SVM Classifier', zxc)
```

We tried to plot a graph for SVM .The code snippet for the same is :

```
X.shape[1]
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))

plt.subplot(1, 1, 1)
plt.subplots_adjust(wspace=0.4, hspace=0.4)

Z = clfs.predict(np.c_[xx.ravel(), yy.ravel()])

# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, cmap=plt.cm.coolwarm, alpha=0.8)

# Plot also the training points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.coolwarm)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xticks(())
plt.yticks(())
plt.title('SVM')
plt.show()
```

The advantages of support vector machines are:

Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.

Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

If the number of features is much greater than the number of samples, the method is likely to give poor performances.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

4. Neural Networks

Neural networks have been successfully applied in a wide range of supervised and unsupervised learning applications. Neural-network methods are not commonly used for data-mining tasks, however, because they often produce incomprehensible models and require long training times. In this article, we describe neural-network learning algorithms that are able to produce comprehensible models, and that do not require excessive training times.Neural Networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks. Biological neural networks have interconnected neurons with dendrites that receive inputs, then based on these inputs they produce an output signal through an axon to another neuron. We will try to mimic this process through the use of Artificial Neural Networks (ANN), which we will just refer to as neural networks from now on. The process of creating a neural network begins with the most basic form, a single perceptron.

Anatomy Of Neural Network

1. Neural Network map a set of input nodes to a set of output nodes.
2. Number of Input Nodes and Output Nodes is variable.
3. The Network itself is composed of arbitrary number of nodes with arbitrary topology.

The neural network used in paper is used to extract features by requiring the network to learn to recreate the input data at the output nodes by using different number of hidden nodes.
A network can be trained to map input values to corresponding output values by providing a training set. The network is repeatedly tested and modified to produce the correct output.
The generation of output by a neural network is accomplished via firing values from nodes. An input is passed to the input layer which in turn can activate the internal layers, which in turn activates the output layer, finally resulting in an output.

Neural Network follows the topology of Back Propagated Network in which inputs are put through a ‘hidden layer ‘before the Output Layer.All the nodes are connected between the layers.

The supervised training of Back Propagated Network includes :
1. Desired output of training inputs.
2. Error= Difference Between Desired and actual Output.
3. Change Weight for more accuracy.
4. Changing the output layer by propagating back to previous layer.
5. Hidden Layers and Number of neurons in the network.

Training the model

Now it is time to train our model. SciKit Learn makes this incredibly easy, by using estimator objects. In this case we will import our estimator (the Multi-Layer Perceptron Classifier model) from the neural_network library of SciKit-Learn.

```
from sklearn.neural_network import MLPClassifier
```

Next we create an instance of the model, there are a lot of parameters you can choose to define and customize here, we will only define the hidden_layer_sizes. For this parameter you pass in a tuple consisting of the number of neurons you want at each layer, where the nth entry in the tuple represents the number of neurons in the nth layer of the MLP model.

```
mlp = MLPClassifier(hidden_layer_sizes=(6,6,6))
```

Now that the model has been made we can fit the training data to our model, remember that this data has already been processed and scaled:

```
mlp.fit(X_train,y_train)
```

Predictions and Evaluation

Now that we have a model it is time to us0e it to get predictions! We can do this simply with the predict() method off of our fitted model:

```
predictions = mlp.predict(X_test)
```

Now we can use SciKit-Learn's built in metrics such as a classification report and confusion matrix to evaluate how well our model performed:

```
cnf_matrix =confusion_matrix(y_test,predictions)
print ("Accuracy of MLP Neural network",accuracy_score(y_test,predictions))
print ("")
print(classification_report(y_test,predictions))
```

To plot the confusion Matrix, a function was made to plot the array with visualization:

```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
```

Information Theory

Information theory measure information in bits.

Information gain=(Entropy of distribution before the split)–(entropy of distribution after it)

Information gain is the amount of information that's gained by knowing the value of the attribute, which is the entropy of the distribution before the split minus the entropy of the distribution after it. The largest information gain is equivalent to the smallest entropy.
The entropy (very common in Information Theory) characterizes the (im)purity of an arbitrary collection of examples.

Entropy Calculations

If we have a set with k different values in it, we can calculate the entropy as follows:
Entropy(p1,p2,…,pn)=−p1log(p1)−p2log(p2)− −pnlog(pn)
If all instances in a group were known to be all in the same class, then the information value of being told the class of a particular instance is Zero

If instances are evenly split between classes, then the information value of being told the class of a particular instance is Maximized.

Information theory measures the value of information using "***entropy***", which is measured in "***bits***".

For example, if we assume in our Occupancy Dataset and take a single attribute Temperature,let's take the dataset between 19 and 21 then let's find its Entropy and Information Gain :

Child Entropy= *-12083/12568 [log(12083/12568)/log 2] - 485/12568 [log(485/12568)/log2]*

=0.235789942

Now, let's find the weight of occupancy dataset who has temperature between 19 and 21 as follows :

Weight of each value = *-P(Customers with Occupancy Temperature between 19 and 21) * [log P(Customers with Occupancy Temperature between 19 and 21)/log 2]*

*= -12568/20560 * [log (12569/20560)/log 2)*

= 0..434041958

Weighted Entropy = *P(Customers with Occupancy Temperature between 19 and 21) * Child Entropy*

*= 12568/20560 * 0.235789942*

=0.144141641

After all the calculations

Information Gain = *Entropy of all the data - Entropy Of Child Expected Data .*

Here, We found Entropy of all the data was and Conditional Entropy of Temperature set between 19 and 21 is 0.530942508

*So,Information gain for Credit History =0.779836704-0.530942508 =*0.248894*.*

4. APPROACH

We followed the approach of **Cross-Industry Standard Process** for Data Mining (CRISP-DM) used commonly by data mining experts for Occupancy Detection Dataset.

4.A) BUSINESS UNDERSTANDING:

To determine how several natural conditions can affect the occupancy of an office room. If the Occupancy is equal to one then occupied and if it is equal to 0 then No Occupancy status. Analyzing which attribute really contributes to the office room. Importing Data from .txt files into Jupyter notebook and concatenating to increase dataset rows. Especially for Neural network where increased no. of training dataset is required to train the machine in regards to supervised learning.

4.B) DATA UNDERSTANDING:

The dataset consists of 20560 readings with attributes like Temperature, Date, Relative Humidity, Light, CO2 , Humidity Ratio and Occupancy.

Attribute:-
date *time year-month-day hour:minute:second*

***Temperature**, in Celsius*
***Relative Humidity**, %*
***Light**, in Lux*
***CO2**, in ppm*
***Humidity Ratio**, Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air*
***Occupancy**, 0 or 1, 0 for not occupied, 1 for occupied status*

The dataset was created from readings and findings found for 9 days between 2nd and 18th Feb of 2015. So day is not a good variable to determine occupancy status. When there is no daylight almost always the occupancy is 0.

4.C) DATA PREPARATION:

The dataset would be subjected for a random state of 0.1 to split the data. Training dataset would be 18504 rows and test dataset would be 2056 rows. Any abnormalities in the data has been cleaned out. Out of 20 attributes Date time stamp is not very useful in regards of predicting occupancy status, therefore this attribute would be removed from further classification. No abnormalities were found in other attributes. Pair plots were drawn between attributes to find patterns. The predicted values are binary and not continuous.

Out[1]:

	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
1	23.18	27.2720	426.0	721.25	0.004793	1
2	23.15	27.2675	429.5	714.00	0.004783	1
3	23.15	27.2450	426.0	713.50	0.004779	1
4	23.15	27.2000	426.0	708.25	0.004772	1
5	23.10	27.2000	426.0	704.50	0.004757	1

Data would look like the above table for most part of the modelling.

4.D) MODELLING:

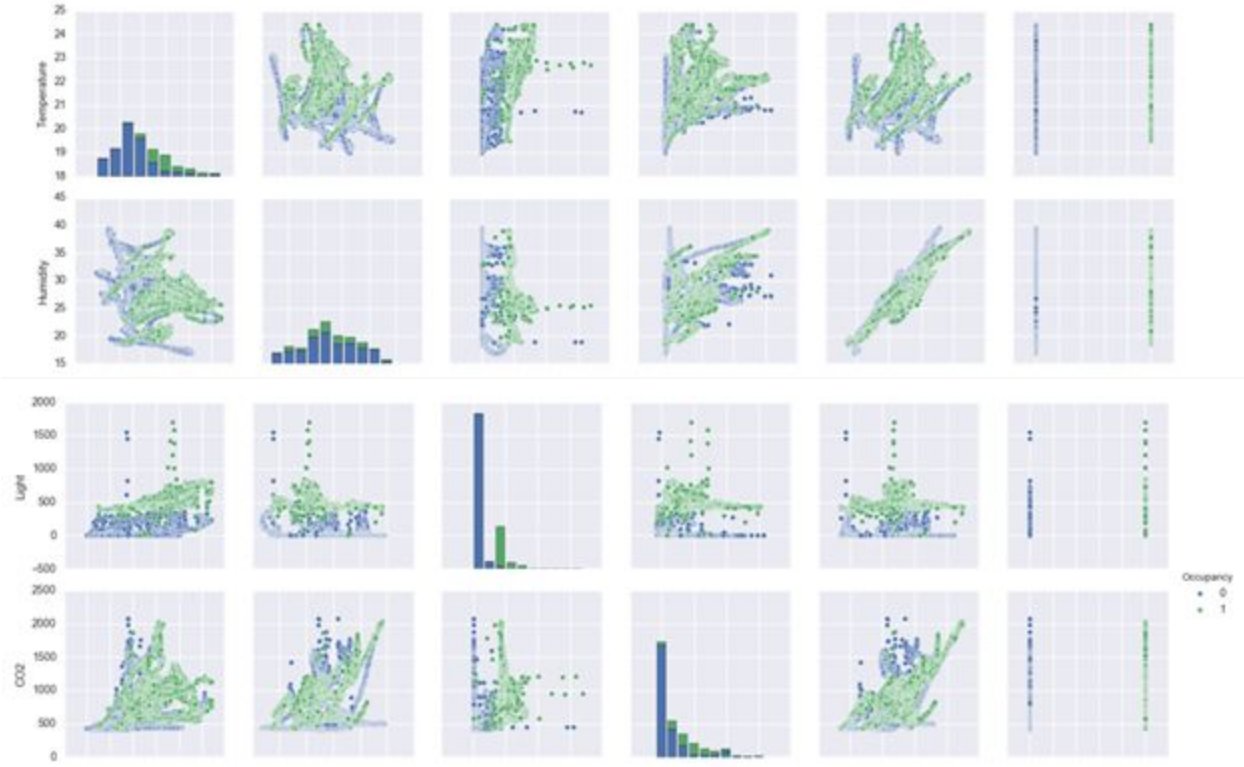
Support Machine Vectors constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks.Decision Trees a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub populations) based on most significant splitter / differentiator in input variables ,Neural networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks.

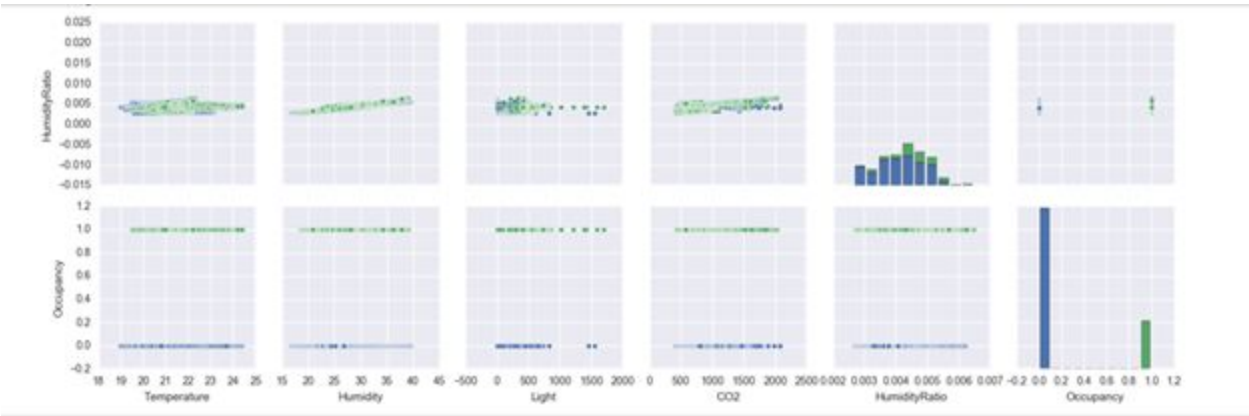
In supervised data mining tasks such as classification, it is common to use error rates as quality measures for data mining models. Therefore, we typically separate the dataset into train and test sets, build the model on the train set, and estimate its quality on the separate test set.The original Datasets were converted to Data training sets, testing sets. A primary component of the plan is determining how to divide the available dataset into training, test datasets.

Run the modeling tool like decision trees using scikit on the prepared dataset to create one or more models. For Decision tree classification, a decision tree figure was generated. The max depth was kept at 3, even 4 was found favorable anything else made the tree complicated. ROC was drawn for Neural Network classifier. SVM plot was drawn for the SVM classifier. Finally Confusion Matrixs were drawn for all 3.

4.E) TESTING AND EVALUATION:

Hidden layers were kept at 6 to compare with 6 tuples (6 attributes) and so as not to complicate the model.





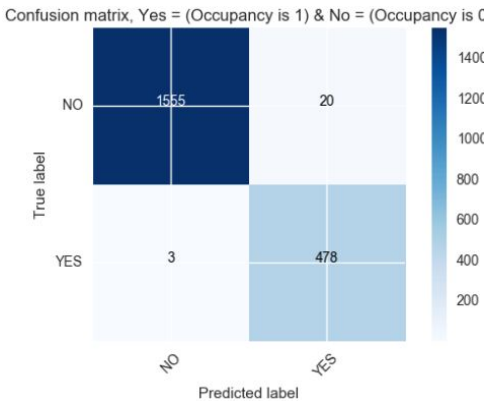


Fig a1. Confusion Matrix - DT

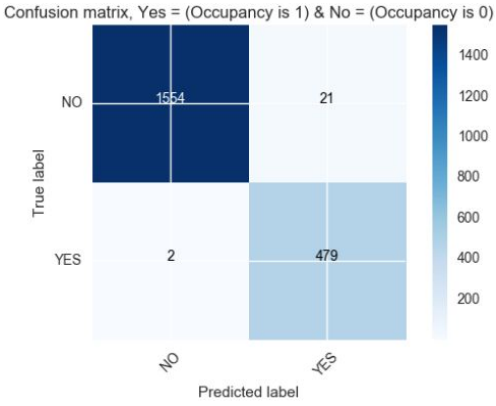


Fig a2. Confusion Matrix - NN

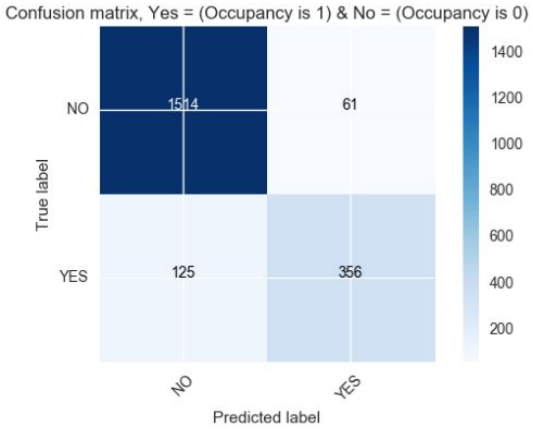


Fig a3. Confusion Matrix - SVM

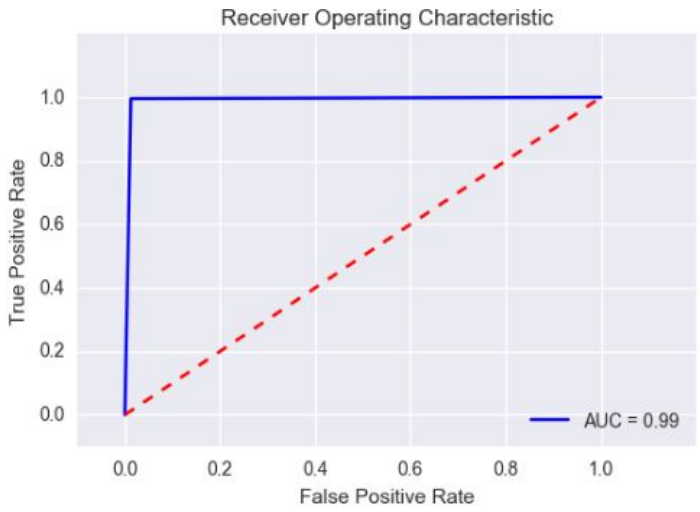


Fig b. ROC

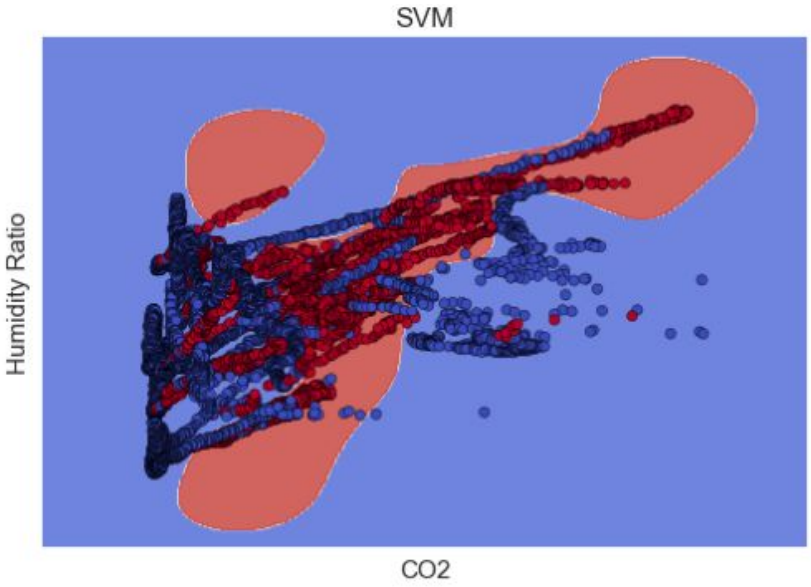


Fig c. SVM

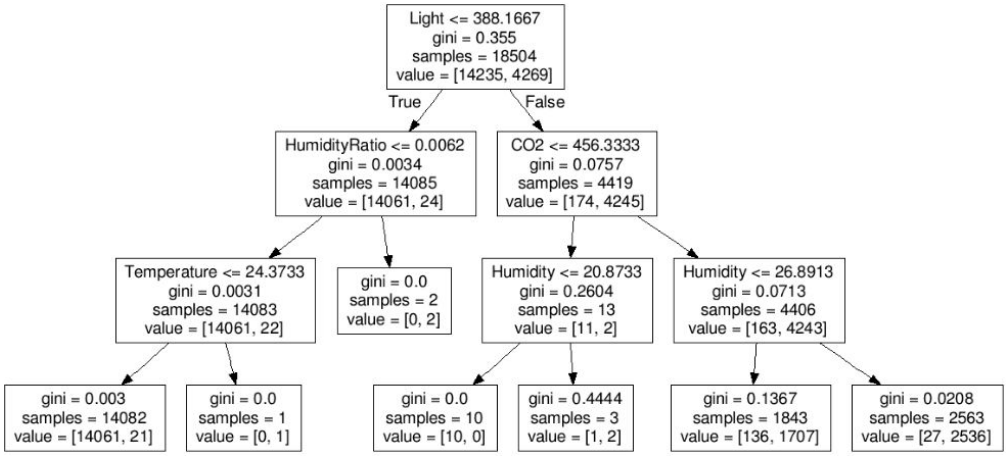


Fig d. Decision tree

Accuracy of Decision Tree = 0.988813229572.

Accuracy of MLP Neural network = 0.988326848249

	precision	recall	f1-score	support
0	1.00	0.99	0.99	1575
1	0.96	0.99	0.98	481
avg / total	0.99	0.99	0.99	2056

Accuracy of SVM Classifier = 0.90953307393

4.F) DEPLOYMENT:

The results show highly positive results for Decision tree and Neural Network both having close to 99% accuracy score. Hence highly reliable to predict occupancy status for the office rooms. Supporting Neural network's claim is the ROC that gives an outstanding area under the curve. And SVM classifier, with accuracy close to 91%, produced lot of outliers as seen in Fig C. Due to this inconsistencies SVM may not be the best choice for the dataset. Probably collecting more data over an year would produce better results.

5. RESULTS AND DESCRIPTION OF RESULTS

The experiments were conducted using the Occupancy Detection Dataset with the help of Decision Tree, Support Vector Machine, Neural network training. Light attribute seems to be the most influential as it tops the node in the Decision tree (Fig d.).

The last column on the classification is Ground-truth occupancy .Thus the prediction is to be done to classify each pattern good or bad.

No.	Attribute	$\mathscr{G}$	$\mathscr{G}'$
1.	Temperature	0.248894	0.210766
2.	Humidity	0.011974	0.006426

3.	Light	0.090242	0.0307446
4.	CO2	0.014051	0.134037
5.	Humidity Ratio	0.252705	0.261335
Average		**0.307562**	**0.12866172**

Table 1. 5 Attribute gains of the Occupancy data set.

Accuracy of 99% achieved for Neural Network is very encouraging, this classifier along with Decision tree would be a great fit for the Occupancy Detection dataset.

References:-

https://biesnecker.com/2015/03/27/visualizing-a-scikit-learn-decision-tree/
http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html#sphx-glr-auto-examples-model-selection-plot-confusion-matrix-py