A

Dissertation On (Major Project-II)

"Analysis of Classification Algorithms"

Submitted in Partial Fulfillment of the Requirement
For the Award of Degree of

Master of Technology

In

Software Technology

By

Jaideep Kumar Vishwakarma
University Roll No. 2K15/SWT/510

Under the Esteemed Guidance of

Prof. Rajni Jindal
Professor& Head of Department, Department of Computer Science & Engineering

2015-2019

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
DELHI – 110042, INDIA

STUDENT UNDERTAKING

Delhi Technological University
(Government of Delhi NCR)
Bawana Road, Delhi- 110042

This is to certify that the thesis entitled **"Analysis of Classification Algorithms"** done by me for the Major project-II for the achievement of **Master of Technology** Degree in **Software Technology** in the **Department of Computer Science & Engineering**, Delhi Technological University, Delhi is an authentic work carried out by me under the guidance of Prof. Rajni Jindal.

Signature:
Student Name
Jaideep Kumar Vishwakarma
2K15/SWT/510

Above Statement given by Student is Correct.

Project Guide:
Prof. Rajni Jindal
Professor & Head of Department, Department of Computer Science & Engineering, DTU

ACKNOWLEDGEMENT

I would like to express sincere thanks and respect towards my guide **Prof. Rajni Jindal, Professor & Head of Department, Department of Computer Science & Engineering, Delhi Technological University Delhi.**

I consider myself very fortunate to get the opportunity for work with her and for the guidance I have received from her, while working on this project. Without her support and timely guidance, the completion of the project would have seemed a far. Special thanks for not only providing me necessary project information but also teaching the proper style and techniques of documentation and presentation.

JAIDEEP KUMAR VISHWAKARMA
M.Tech (Software Technology)
2K15/SWT/510

ABSTRACT

In this, I have analyzed and compared different classification algorithms and run on dataset and compared efficiency, and using combination of this algorithm shown to improve efficiency. I have used and trained model with four different classification algorithm and used those model to predict outcome of test data separately and then used combination of those model to improve production efficiency of test data prediction.

With the advent of technology and machine learning development, there are various algorithms developed for model to be prepared which can take decision and predict something based on its learning.

Many of research and analysis are based on these classification algorithms. It plays important role in machine learning. It helps to create model which can predict outcome based on its past learning. So if a model has multiple algorithms as a factor to decide or predict outcome it will be having good efficiency in order to predict right.

As various classification algorithms exists so in order to pick right or suggest using combination of these algorithms in order to improve prediction efficiency. There are always a tradeoff between efficiency and execution time, here in this we have majorly focus on improving efficiency by using existing algorithms for a single model which consist of different models with different algorithms.

Here in this I have majorly used skit-learn[1] libraries interface for various classification algorithms. I have used them in order to check and show efficiency of predicted output. By efficiency I mean with respect to correct output prediction. I have used various trained model for one single model in order to improve efficiency.

TABLE OF CONTENTS

CHAPTER 5

CHAPTER 6

Chapter 1: Introduction

1.1 PROBLEM STATEMENT

In current world technologies are improving a lot, every day there are new invention and discoveries. In the same way requirement are also increasing, automation, machine learning, Artificial Intelligence, robotics etc are the current world topics in which more focus is given nowadays.

Various companies are working towards Artificial Intelligence and machine learning and keep improving things in various aspects. Few example of such learning applied by various companies like Netflix applies these in order to suggest recommendation for their TV shows/movies to customer based on their profile and learning from so many other profiles, finding similarity and suggestion content to be watched.

This work is to provide complete guide to students and some professionals to use machine learning specially classification algorithms and use them to do multiple tasks. This work will help to list down criteria to pick some algorithm and suggest to use multiple algorithms in order to improve efficiency for work that doesn't have much time constraints.

In this work analysis of classification algorithms are done in order to solve classification problem and shown that in order to improve efficiency combination of these algorithms can be used for not time constraint task like analysis and classification of user interest and match with different user's interest and produce a recommended system. It will help other students or professional to understand classification in machine learning terminology.

1.2 WHAT IS MACHINE LEARNING

Machine learning is an application of artificial Intelligence that provides system ability to learn and improve itself without programming explicitly. Machine learning makes a system capable enable to take decision and improve its self with its own experience. There are mainly three ways for system to learn.

1.2.1 SUPERVISE MACHINE LEARNING

This method of machine learning, is the learning with labeled data, to learn labeled data of past and predict future, In this method learning is there to form some inferred function or mechanism to predict outcome and correct itself using the correct label study of that outcome.

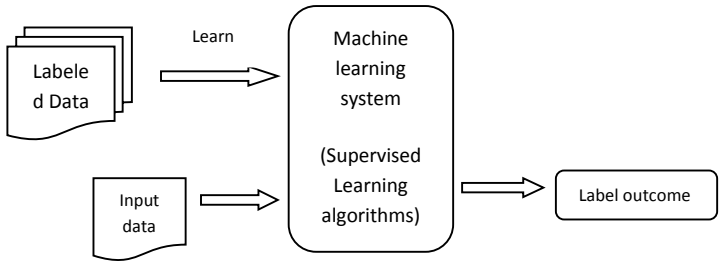


Fig 1.1: supervise learning

1.2.2 UNSUPERVISE MACHINE LEARNING

This method of machine learning works with unlabeled and unclassified data. This method used algorithm to analyze given data and try to identify hidden structure and classify them based on common attributes if required. It might not give right output but it infers the hidden structure and compares testing data with it to find conclusion.

In this method algorithms analyze data and find its underlying structure and keep them in group and classify them. This is used when we want to find some hidden pattern or structure in the given data. We can correct its output make them learn and adjust its finding of structure of data.

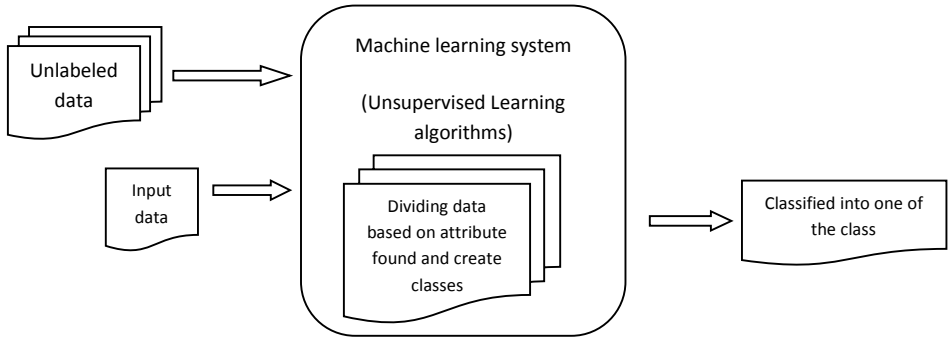


Fig 1.2: supervise learning

1.2.3 REINFORCED MACHINE LEARNING

This method of machine learning involve concept of reward and punishment, for every correct prediction it gives system some reward and for every wrong prediction it gives some punishment. Based on this reward and punishment, system learns and improves itself.

In this algorithms need to find balance between current knowledge and knowledge it will get by exploring given data.

1.3 THESIS MOTIVATION AND GOAL

Machine learning, a part of Artificial Intelligence, has now current world most promising systems for future. To give machine intelligence to decide its own logic based on the facts and data it has.

One such important part is classification, these systems must understand and equipped with classification on the basis of attributes of tons of data.

One of major task in machine learning is classification and clustering. Classification means arranging data into classes of similar attributes. For example Apple and Orange, they have man similar attributes like both are approx round; both are fruits, juicy etc. even though they are different.

So classification is basic part for all these machine learning, which we have explored in thesis, there are many algorithms for classification, we will look few important algorithms and analyze those algorithms and see how using combination of many algorithmic model might give high efficiency.

Many times our requirement is not to give fast result but more efficient result, so I checked and analyze these algorithms and found that if we use combination of such algorithm then we can have better result, although time taking to give this result is high. It will help to understand and analyze these algorithms and how to improve efficiency by combining many of such models.

1.4 THESIS ORGANIZATION

This thesis is classified into six different chapters.

Chapter 1 defines the problem statement for the thesis, which is analysis of Classification Algorithms. Prerequisites have been explained there. Motivation and goal is also mentioned.

Chapter 2 describes the related work done in the areas of analysis of Classification Algorithms and use of those algorithms.

Chapter 3 explains the topics used in detail. In our research details, we explain the terms, which are being used in the thesis like different Classification algorithm. Different terms involved in understanding of these algorithms.

Chapter 4 explains architecture of proposed system which utilizes the internal knowledge of these classification algorithms and model based on these algorithms. Describe the proposed model and compare efficiency in terms of correct classification of test data. Also contains proposed approach and code snippet and description about data being used for classification test.

Chapter 5 explains about Result and analysis done.

Chapter 6 gives the conclusion about my complete work done and gives few points for future work.

Chapter 2: Related Work

[1] Describe and provide libraries to use default implementation of various algorithms used for classification and other things.

[2] Describe and provide library api design for various algorithms and given parameter for modification.

[3] Describe to use kNN algorithm one of classification algorithm use to classify data.

[4] Describe about Decision Tree Algorithm for classification and its application in classification.

[5] Describe and does comparative study of various classification algorithms.

Chapter 3: Research Background

3.1 Classification

Classification is technique of distributing set of data into classes with similar properties. In other words classification is the process of labeling some data based on its internal structure and properties. In classification process there are few defined classes or categories are there, process of classification will iterate each data and put a class/category label on each. There are many classification algorithms present in current world and there are many optimization also proposed on those algorithms. We will define few of them and then analyze combination of these algorithmic models in order to improve efficiency.

These algorithms are being used in machine learning basically to label, unlabeled data based on its past learning and understanding. Some of the algorithms also include dimensionality reduction mechanism which includes removal of attributes which don't contribute much in order to predict in the process of classification.

3.2 kNN(k-nearest neighbor) Algorithm:

kNN algorithm is very basic algorithm in classification task. It's kind of extension of nearest neighbor algorithm. It is based on the principal that what is the nearest neighbor class for the data being test. It calculates distance from given data and try to find out where this particular data lies.

It uses complete data which is given when trying to predict label or class of test data. So as such model developed with this algorithm don't have learning only it has its strong data which it will use to predict about test data. As this uses full data to predict so efficient data structure should be use to store this data and add or adjust data when some new data comes. To make sure all data which is being used is proper so that data can be purified or preprocessed to remove outliner data etc.

To predict new data instances (x) it tries to find in all data for K similar data instances and then label this new instance with the same label. To determine which K instances are more similar from the data base it measures Euclidean distance [7].
Euclidean distance between two instance /points P and Q, which have attributes distributed in an n-dimensional space $(p_1,p_2, p_3....p_n)$ $(q_1,q_2,q_3....q_n)$ is defined as below.

$$\text{Euclidean distance } D(P,Q) = \sqrt{\sum_i^n (p_i - q_i)^2}$$

This algorithm calculates distance of given test data from all the data instances given for learning, then according to value of *k* it will check k similar data instances which have minimal distance to the test data and assign same label to test data which those similar instances has.

Suppose trained data instances are $I_1, I_2,....., I_n$. and corresponding label of these instances are L_1, $L_2,.....L_n$, and test data instance which we need to check is T then Euclidean distance from all given data instances $D_1(T, I_1)$, $D_2(T, I_2),.....$ $D_n(T, I_n)$, sort them in ascending order, then choose k instances whose label is same and assign test instance T as same label L.

3.3 Decision Tree Algorithm:

Decision Tree Algorithm is most popular algorithm in Machine Learning as it mimics human brain behavior. Like human brain it create choice or option and create correct question to ask, for example "Is it raining today?" then "need to carry Umbrella" something like this. This algorithm creates question nodes at each level and move from root node to leaf to find or classify test instance.

Decision Tree algorithm creates tree of right questions or conditions following which from top to bottom a test instance can be given label. Main question is how this algorithm knows what to ask or say how to decide root node. This is also used in case of binary classification.

To create and identify root node, it first analyze problem mathematically. By the use of Information Theory, it calculates Entropy and Information gain with various attribute of learned data instances and it creates node with which we get maximum information gain. This method of creation of tree is also known as ID3 (Iterative Dichotomiser 3). To understand, let's first understand Entropy and Information Gain.

3.3.1 Entropy

Entropy of data instances given is basically measure of homogeneity of data. If sample data is homogenous then entropy is 0 and if it's equally divided then its entropy is 1. Entropy function can be defined as below in terms of frequency of occurrences of attribute values in data instances. For binary classification, If frequency of an attribute in terms of probability is p_i then Entropy E(S) is defined as:

$$E(S) = \sum_i^n -p_i \, log_2 p_i$$

Sum of all entropy for different attributes with their occurrence probability gives complete entropy of that attribute. All attribute's entropy will be used to calculate information gain. And then root node is created based on information gain.

3.3.2 Information Gain

Change in entropy is known as Information gain. It's based on simple principle of how much information can it retrieve by using a particular attribute with its values. If change in entropy is 0 then it means there is no information gain so that particular attribute and value will not be selected for node.

Information Gain G(T, X) = Entropy (T) – Entropy(T, X).

Here Entropy (T) is overall entropy and Entropy (T, X) is entropy for a attribute with all its possible values. So whatever gives more information gain means more information towards identifying or categorizing data instances will be used as root node. Same thing then be used to generate other tree nodes until tree completes.

Using this information gain, Decision tree is made and model is prepared with this decision tree on trained dataset. So when new test data instance come then it traverse this decision tree in order to classify test instance. In this way new instance of data can be classify.

3.4 Support Vector Machine:

This algorithm is also widely used in supervised machine learning; it is being use for both classification and regression. But our use will be to classify test data. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

The idea behind this algorithm is to find a hyper plane between 2 different classes in order to separate instances of these 2 classes having maximum margin. What we understanding from maximum margin is that for all vectors of these classes must have maximum margin from this hyper plane. In order to predict correct class for test instance this margin is very important.

Let's consider one example and see what we mean for maximum margin and hyperplane. Suppose 2 classes are defined as show in below figure.

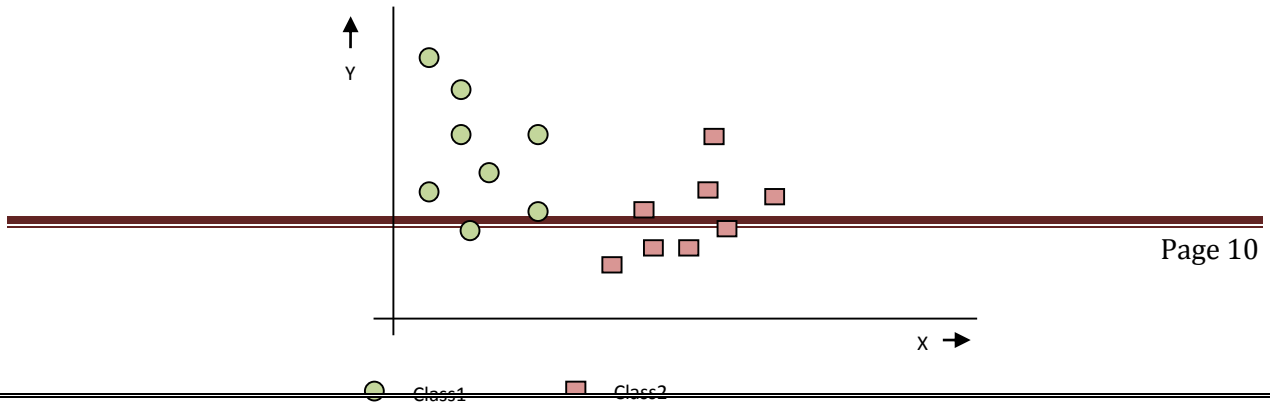


Fig 3.4.1: Data plot for SVM data

To understand fully, let's assume we have one test input of class 1, for which we need to predict its class using SVM algorithm prediction. There can be many planes in order to separate these 2 classes as shown.

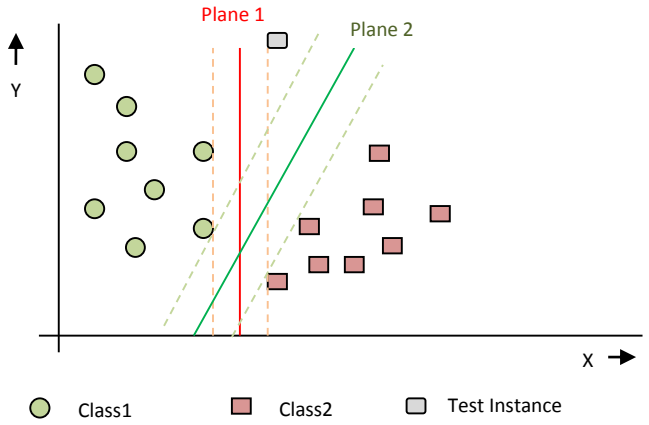


Fig 3.4.2:2 of possible Hyper planes

In above fig we have shown 2 of many possible planes in order to separate these classes. Plane 1 have not maximum margin from these classes instance. Plane 2 however have maximum distance. A SVM algorithm tries to find this hyper plane which has maximum distance from all of data instances points. So it solves optimization problem and once it find plane with maximum distance it stops and create barrier between these 2 classes.

Now if we see for test Instance to find in which class it belongs. If we would have considered plane1, then it would have classified wrong. But plane 2 is classifying this data instance properly as its belong to class 1.

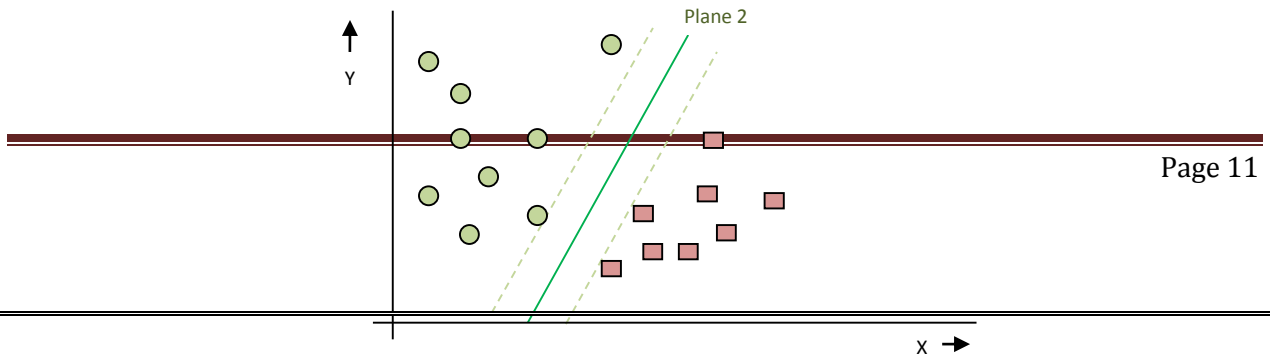


Fig 3.4.3:Correct prediction of test instance

There are few tuning parameters use to tune algorithm as per requirement. One of them is called Kernel. Kernel is responsible to change data dimensionality. For example if it's not proper to find hyper plane (a line) in 2 dimensional data then it convert these data into 3-d and then find a hyper plane (a plane).

Another parameter is regularization parameter (C), which is defined in sklearn SVM algorithm to control how much it's okay for wrong prediction. Higher the value of this parameter have lower wrong prediction and for this it finds many connected hyper plane in order to separate data classes.

Chapter 4: Proposed Approach

4.1 Architecture for New Model

New model what we will develop is inspired from and use kNN, Decision Tree and SVM algorithms in order to classify data. The idea behind this model is to use already available tools

and algorithms and use them in a fashion to get increased efficiency. It is similar to hybrid approach for creating new model. Combination of using these existing models into one model will improve efficiency. It will result an output from individual model output and choose output which are same for more model.

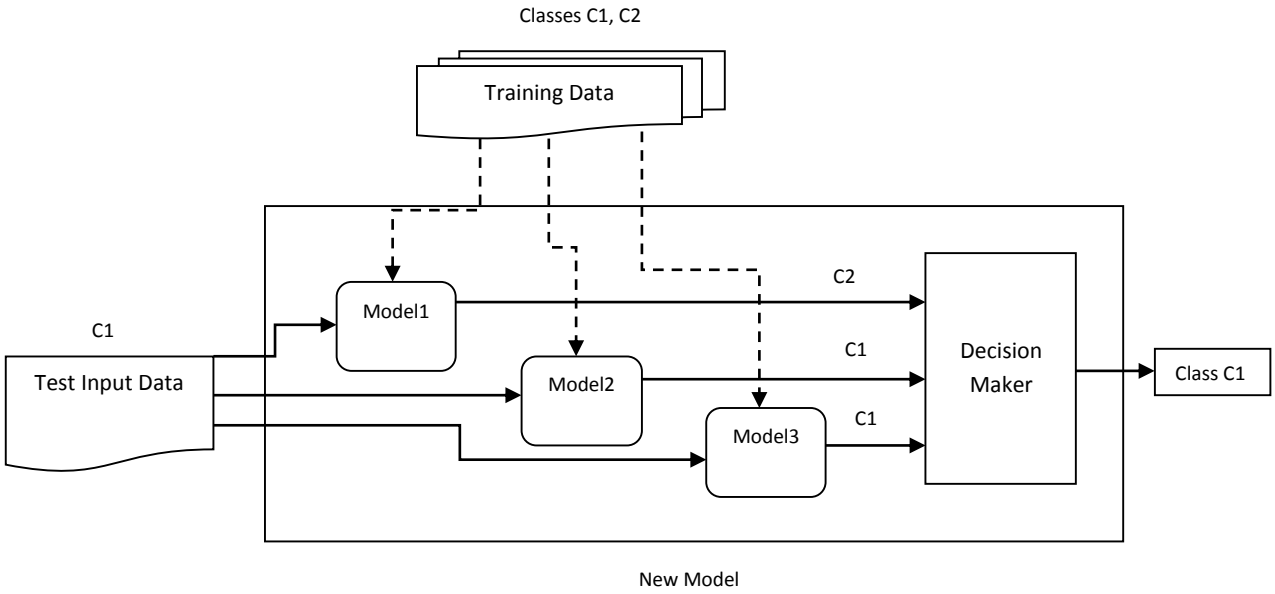


Fig 4.1.1:Design for Proposed new Model

As shown in above figure, there are 3 models, Model 1, Model 2 and Model 3, all these model received same training data, but uses different algorithm. Model 1 uses k-NN algorithm, Model 2 uses Decision Tree and Model 3 uses SVM algorithm in order to classify given data. Suppose given training data have two classes C1 and C2. This training data is given to all models Model 1, Model 2, and Model 3. After training one input data comes to find out in which class it belongs. Every model uses its own algorithm logic in order to predict class for given test input data. Every model gives output based on their own algorithm. These output data is then passed to a decision maker module which will check and all model output and try to predict correct output class. This way it will improve efficiency as it will not consider only one model output but it will consider all 3 model output and then decide and pick the correct one.

Training Data: Training data is the data instances which are provided to these models in order to train themselves understand and develop internal model in order to predict outcome on similar inputs. Training data is use to train these models, as much as this data is present, chances are model will be trained better as it will understand all data and create logic. This training data must not be having outlier values, so sometimes preprocessing is required to remove data which are very much deflective from the rest of the data, because if these data are present then underline algorithm will take these into account and may create not very efficient model.

Model 1 (k-NN Algorithm): Model 1 is the model using k-NN algorithm, it uses k-NN algorithm and train on the training dataset and ready to predict the outcome of test input data instance. k-NN algorithm use full dataset in order to find the location of new data instance need to be classified. There are various implementations and version available for k-NN algorithm which tries to reduce the data usage and store the components from which new instance location can be identified. Every new instance will make corresponding structure to be modified, If included in learning.

Model 2 (Decision Tree Algorithm): Model 2 is the model using Decision Tree algorithm, it uses Decision Tree algorithm to train and predict the outcome of supplied test data instances. This algorithm is based on Information theory coding principles and it uses them in order to create a decision tree. Its set of questions, answers of which leads to a branch of that tree where you find another question, move this way till the time you reach leaf node and identify class of given test data. This model have high no of correct prediction as it uses Information theory principles and mathematically calculate the actual class of the given test data. Based on the data and parameter tweaks may lead to higher efficiency.

Model 3 (Support Vector Machine Algorithm)): Model 3 is using Support Vector machine algorithm. It uses this algorithm to train and predict outcome of the test data instances. It also uses mathematical principles in order to create model for logic to predict or classify test data instance. It is most widely use algorithm as it not only create straight planes also it can classify data and make boundaries around them in any of the structure be it circle or any arbitrary structure.

Decision Maker: This part of the proposed system is responsible to accept output of all three models and decide what should be actual output. This decision maker part can easily be modified with various techniques. Decision maker first check if 2 models are giving same output if it is same then that is the output of the proposed system. Also it gives points to the model used in terms of probability, for each of such situation both model having same output been given 0.4 points to each and remaining 0.2 to last machine, It has been done in order to take wrong answer selection into account. If all model is saying different outcome then it takes the output of model having highest collected points and rewards equal points (0.33) to each model which will be used in future prediction. For example consider below table to understand awarded points. In below table all three models have output mentioned for three test instances along with points awarded to them. So if case happens like in test instance 3 then the output selected will be of model 2 as it has total points higher than the rest of the models. But all will be awarded with equal points as none of them given certainty of information that this is more probable output.

Models	Test 1 Output	Points Awarded	Test 2 Output	Points Awarded	Test 3 Output	Points Awarded	Total Points
Model 1	C2	0.2	C1	0.4	C1	0.33	0.93
Model 2	C1	0.4	C1	0.4	C2	0.33	1.13
Model 3	C1	0.4	C2	0.2	C3	0.33	0.93

Table 4.1.1: Example values for proposed System.

Using these values for future prediction and choosing the correct output based on the higher points. It's made them similar to reinforced learning but still even though they get points it's not sure that which model is giving correct output.

As multiple algorithms is being used in order to classify data. So efficiency will be little higher as it will pick the best outcome based on historic data and learning. These model will use their learning in order to predict/classify data. But this complete model will use these model learnings along with historic data and classify input data.

Efficiency: Efficiency of any classification model can be defined as the probability of correct classification from total no of output classification. Suppose a model is being tested with N test data instances and this model is able to correctly classify only C items where C <=N. Then efficiency of this model can be represented as

$$\text{Efficiency E} = \frac{C}{N}$$

I have used sckit-learn framework for existing model algorithms and use their combine output for input of Decision maker module, which interns gives final output.

Efficiency of the proposed machine is highly dependent on individual machine performance and how likely two machine choose the wrong result. That's why I have chosen these 3 algorithms, Support Vector Machine, KNN algorithm, Decision Tree algorithm.

Decision maker can be modified in other way also to give points on the likelihood of giving wrong answer, and accordingly create table and maintain history and then select output which shows minimum wrong answer calculations.

Implementation Details:

IDE/Framework Used – PyCharm
Library Used : sckit-learn, numpy, matplotlib
Language Used – Python

Dataset Used : http://mlg.ucd.ie/datasets/bbc.html
BBC raw data files

Model 1: K-NN Classification is being used, We have used 80% data for training and remaining 20% is for test run.

Parameters we have used are as below.

X_train: Having the data matrix parameter, used for Training.

Y_train: Having the label for corresponding X_train data.

X_test: Having matrix of data which will be used to test.

Y_test: Will have predicted classes label for corresponding X_test data.

classes: Classes are the dictionary of classes in which Machine as has to classify data.

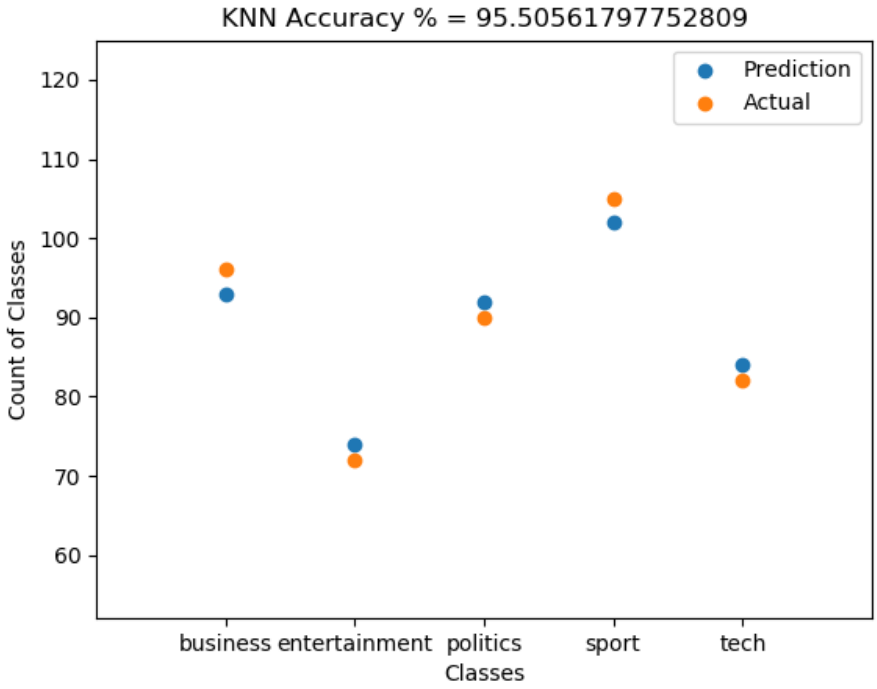
```
def knnMachine(X_train, X_test, Y_train, Y_test, classes):
    knn = KNeighborsClassifier()
    knn.fit(X_train, Y_train)
    predict = knn.predict(X_test)
    score = knn.score(X_test, Y_test)
```

This will use knn classifier from sckit-learn package and train data for the same.

```
#Accuracy/Efficiency
print('KNN Accuracy = ' + str(score))
```

KNN Accuracy =0.9550561797752809

This will print the graph plot with respect to predicted classes count with actual classes counts of test data.



Model 2: Decision Tree classification is being used in this model, I have used 80% data for training and 20% data for test run.

Parameters we have used are as below.

X_train: Having the data matrix parameter, used for Training.

Y_train: Having the label for corresponding X_train data.

X_test: Having matrix of data which will be used to test.

Y_test: Will have predicted classes label for corresponding X_test data.

classes: Classes are the dictionary of classes in which Machine as has to classify data.

```
def decisionTreeMachine(X_train, X_test, Y_train, Y_test, classes):
    decisionTre = DecisionTreeClassifier(criterion="entropy", random_state=10)
```

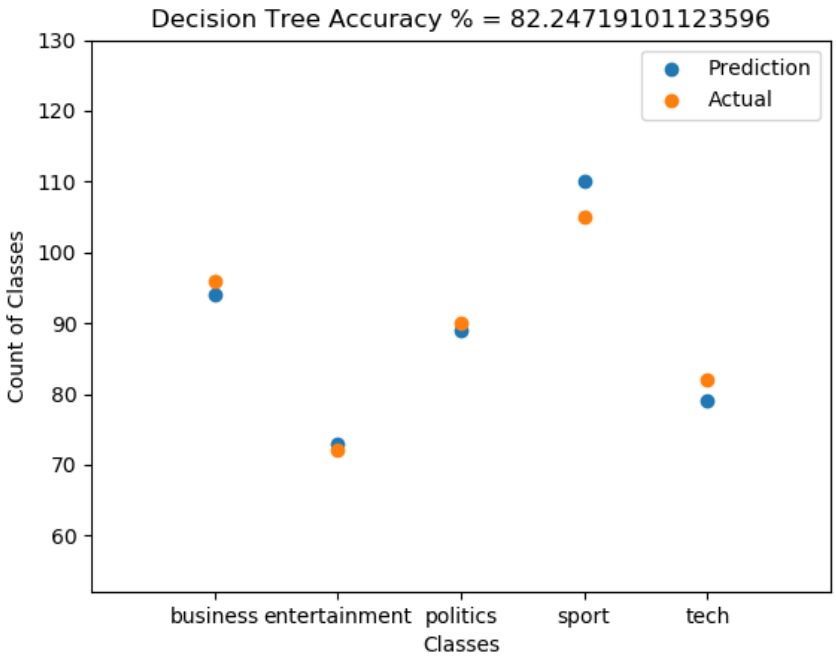
```
decisionTre.fit(X_train, Y_train)
predict = decisionTre.predict(X_test)
score = decisionTre.score(X_test, Y_test)
```

This will use Decision Tree classifier from sckit-learn package and train data for the same and Calculate Efficiency

```
#Accuracy/Efficiency
print('Decision Tree Accuracy = ' + str(score))
```

Decision Tree Accuracy = 0.8224719101123595

Graph plot for Decision Tree classification with respect to predicted and actual data.



Model 3: This model uses Support Vector Machine classification, this also uses 80% of data for training and remaining 20% data for test run.

Parameters we have used are as below.

X_train: Having the data matrix parameter, used for Training.

Y_train: Having the label for corresponding X_train data.

X_test: Having matrix of data which will be used to test.

Y_test: Will have predicted classes label for corresponding X_test data.

classes: Classes are the dictionary of classes in which Machine as has to classify data.

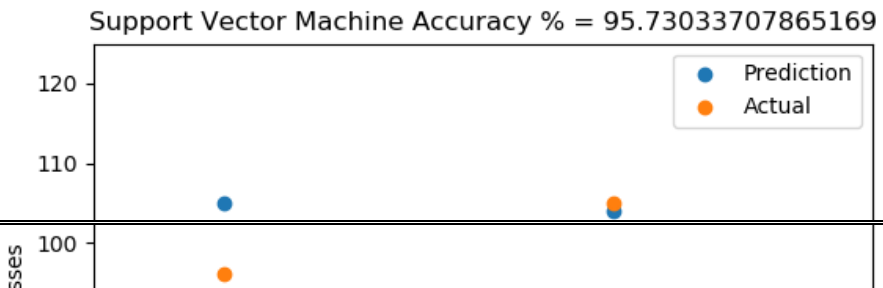
```
def svmMachine(X_train, X_test, Y_train, Y_test, classes):
    svc = NuSVC(gamma="auto")
    svc.fit(X_train, Y_train)
    predict = svc.predict(X_test)
    score = svc.score(X_test, Y_test)
```

This will use NuSVC classification of Support Vector Machine for classification and then I have calculated efficiency.

```
#Accuracy/Efficiency
print('Support Vector Machine Accuracy = ' + str(score))
```

Support Vector Machine Accuracy =0.9573033707865168

Graph plot for this Machine with respect to predicted and actual classes of test data.



Proposed Model: This model uses combination

Parameters we have used are as below.

Y_test: Will have predicted classes label for corresponding X_test data.

classes: Classes are the dictionary of classes in which Machine as has to classify data.

svmPredict: predicted values from svm model

knnPredict : predicated values from knn model

dtPredict: predicted values from decision tree model.

```
def newModel(Y_test, classes, svmPredict, knnPredct, dtPredict):
```

```
count=0
size = np.asarray(svmPredict).size
solArr = np.empty(size, int)

for index in range(size):
    one = svmPredict[index]
    two = knnPredct[index]
    three = dtPredict[index]
    if one==two:
        solArr.put(index, one)
    elif two==three:
        solArr.put(index, two)
    elif three==one:
        solArr.put(index, three)
    else:
        solArr.put(index, one)

solArr = solArr.tolist()
eff = calculateScore(Y_test, solArr)
```

This is using general mechanism if any of the two output from different algorithm is same then it is picking that one and storing it into new array.

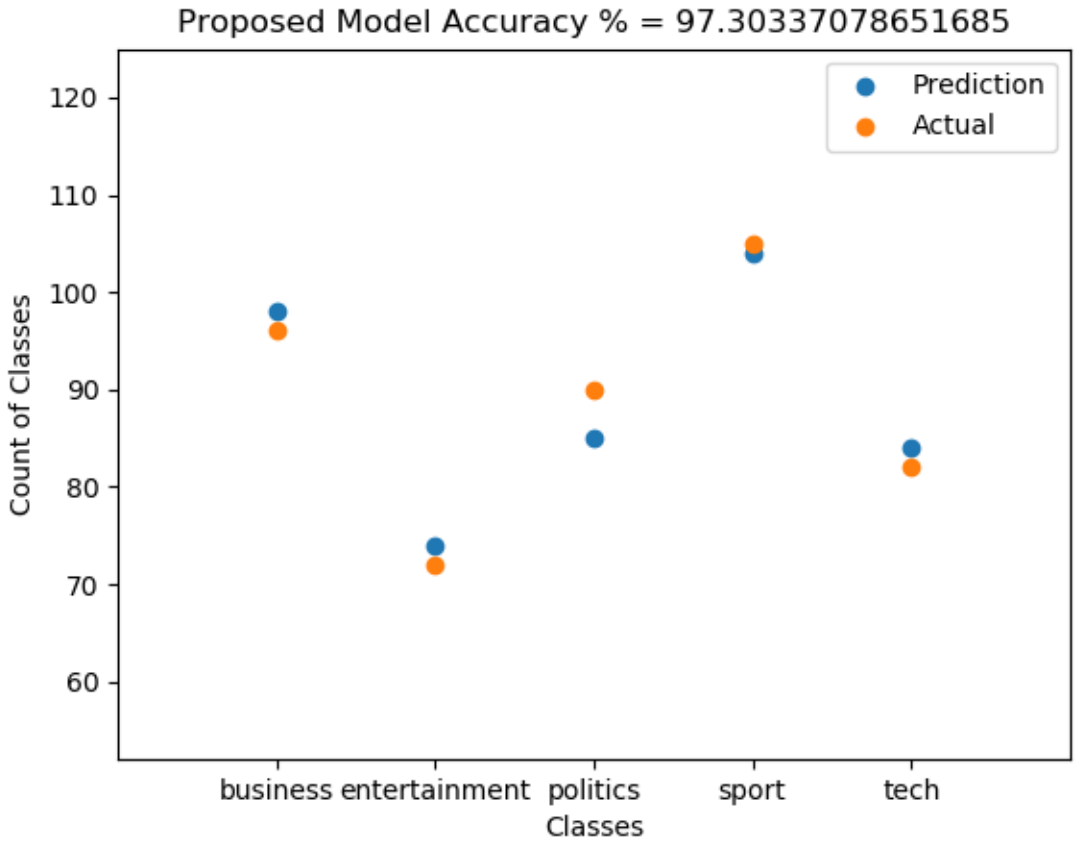
```
def calculateScore(Y_test, Y_predict):
    size = np.array(Y_test).size
    count = 0;
    for i in range(size):
        if Y_predict[i]==Y_test[i]:
            count = count+1

    eff = count/size
    return eff
```

This is used to calculate score of new model and efficiency parameter is returned.

```
#Accuracy/Efficiency
print('New Model Accuracy = ' + str(eff))
```

New Model Accuracy =0.9730337078651685



Efficiency or say accuracy of this proposed model is greater than the existing model shown above.

Chapter 5: Results and Analysis

5.1 Data Collection and Preprocessing Details

Data was collected from BBC, its BBC news database having news divided into 5 categories. It has around 3K data. I have used this data for our existing model study and use the same data in our proposed model. Five categories of this database are “business”, “entertainment”, “politics”, “sports” and “tech”. These are raw data and organized into folder of these five categories. I have read these raw data and read folder name as classes of these categories. I have used 80% data for training and 20% data I am using as test instances for prediction of classes of these data.

These raw data is read and used *‘TfidfVectorizer’* to preprocess this data in English and extract features out of them and create feature matrix. These feature matrixes are being used for analysis and training purpose and some of the data is being used for testing purpose. Proposed model will extract its learning from testing data also, it will build up a model for testing data also so that once real word testing data comes then it can utilize them and keep adding into its learning.

DATA SUMMARY FOR OUR MODEL:

Data	Total Categories	Raw Data each category(avg)	Total Records	Training Data Size (80%)	Testing Data Size (20%)
BBC News	5	445	2225	1780	445

Table1: Data Summary for model

5.2 Implication of Results

Choosing the right decision maker will improve accuracy for predicting data. This model will take more time but with help of decision maker it will improve accuracy of existing individual models. So this result implies that we can use combination of such algorithm with some decision maker in order to select the output for corresponding test input data.

Chapter 6: Conclusion

6.1 Conclusive Results

In this project I have used BBC news data which was categorize into five categories, our model is using set of existing models and one decision maker which decide what to output and what output to choose. We have collected output from existing models and pass it into decision maker and as shown we have improved efficiency in our new model. Improvement of my new model will be based on decision maker algorithm. If we use more pure and more aggressive algorithm we can improve result. But choosing wrong algorithm might drastically decrease the performance result. As shown in the implementation section we have got good result and improved efficiency even using very simple decision making. .

Total Test Data	Total Classes	Total Data set	Accuracy Measure for model of Activity recognition
445	5	2225	97%

Table 2: Data summary for our new model

6.2 Conclusive Summary

The proposal that we have presented contain design and implementation of New Model which uses three existing models and one decision maker to take decision on various inputs. This system can be used to train less no of data and use combination learning to predict right output.

Implemented Model can be run once trained on any system having capability to run native code. It can also use on Android devices if required. Its use is basically limited in the area where time constraint is little relaxed, as it take much space and keep track of data depending on output selection. Result shown which run on BBC news database that this system has improved efficiency then its individual component.

6.3 Future Work

The future task would be to make this less time consuming and generate other algorithms in order to make decision. It can have more no of individual machines and decision make can be written more properly like using probabilities and continuous learning.

REFERENCES

[1] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[2] API design for machine learning software: experiences from the scikit-learn project, Buitinck et al., 2013.

[3] Halil Yigit (2015) ABC-based distance-weighted kNN algorithm, Journal of Experimental & Theoretical Artificial Intelligence, 27:2, 189-198.

[4] Decision tree methods: applications for classification and prediction Shanghai Arch Psychiatry. 2015 Apr 25; 27(2): 130–135.

[5] Sunita B. Aher, Lobo L.M.R.J., "COMPARATIVE STUDY OF CLASSIFICATION ALGORITHMS", International Journal of Information Technology and Knowledge Management, July-December 2012, Volume 5, No. 2, pp. 239-243

project_rj_1

by Malaya Dutta Borah

project_rj_1

ORIGINALITY REPORT

%11	%7	%4	%10
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	louisdl.louislibraries.org Internet Source	%1
2	Submitted to Glasgow Caledonian University Student Paper	%1
3	cso.kmi.open.ac.uk Internet Source	%1
4	ecommons.usask.ca Internet Source	%1
5	Submitted to Delhi Technological University Student Paper	%1
6	ijiset.com Internet Source	%1
7	Guifen Zhao, Yanjun Liu, Wei Zhang, Yiou Wang. "TFIDF based Feature Words Extraction and Topic Modeling for Short Text", Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences - ICMSS 2018, 2018	%1

	Publication	
8	Submitted to The Robert Gordon University Student Paper	<%1
9	Submitted to Higher Education Commission Pakistan Student Paper	<%1
10	Submitted to University of Wales Institute, Cardiff Student Paper	<%1
11	Submitted to Harrisburg University of Science and Technology Student Paper	<%1
12	Submitted to Feng Chia University Student Paper	<%1
13	Submitted to Study Group Australia Student Paper	<%1
14	Submitted to Deakin University Student Paper	<%1
15	Submitted to Northern Consortium UK Student Paper	<%1
16	mathstatisticspython.blogspot.com Internet Source	<%1
17	Submitted to Sim University Student Paper	<%1

18	Submitted to Ohio University Student Paper	<%1
19	Submitted to Universiti Teknologi Petronas Student Paper	<%1
20	www.i-scholar.in Internet Source	<%1
21	Submitted to Auckland University of Technology Student Paper	<%1
22	Submitted to University of Westminster Student Paper	<%1
23	mfr.edp-open.org Internet Source	<%1
24	Submitted to University of Sydney Student Paper	<%1
25	Submitted to Middlesex University Student Paper	<%1
26	irihs.ihs.ac.at Internet Source	<%1
27	pt.scribd.com Internet Source	<%1
28	Submitted to University of Leeds Student Paper	<%1

29	"Computational Intelligence in Data Mining", Springer Science and Business Media LLC, 2017 Publication	<%1
30	logicomatematicoi.blogspot.com Internet Source	<%1
31	rapidminer.com Internet Source	<%1
32	open.library.ubc.ca Internet Source	<%1
33	orca.cf.ac.uk Internet Source	<%1
34	pnrsolution.org Internet Source	<%1
35	Submitted to University of Zakho Student Paper	<%1
36	Submitted to University of Nottingham Student Paper	<%1
37	Submitted to CVC Nigeria Consortium Student Paper	<%1

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE	OFF		

BIBLIOGRAPHY