IDENTIFY BLOOM KNOWLEDGE OF STUDENT

MINI PROJECT REPORT
SUBMITTED TO

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)

Bangalore - 560054

SUBMITTED BY

M Sneha	1MS14CS058
Mipsa Patel	1MS14CS148
Tilak S Naik	1MS14CS134
Vibha Karanth	1MS14CS136

As part of the Course **Data Analytics Laboratory - CSL717**

SUPERVISED BY
Parkavi. A

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAMAIAH INSTITUTE OF TECHNOLOGY

Sep - Dec 2017

Department of Computer Science and Engineering

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Bangalore - 54

CERTIFICATE

This is to certify that M Sneha (1MS14CS058), Mipsa Patel (1MS14CS148), Tilak S Naik (1MS14CS134), Vibha Karanth (1MS14CS136) have completed “Identify Bloom Knowledge of Student” as part of Mini Project.

We declare that the entire content embodied in this B.E. 7th Semester report contents are not copied.

Submitted by

M Sneha	1MS14CS058
Mipsa Patel	1MS14CS148
Tilak S Naik	1MS14CS134
Vibha Karanth	1MS14CS136

(Dept of CSE, RIT)

Guided by

Prof. Parkavi
Assistant Professor
(Dept. of CSE, RIT)

Department of Computer Science and Engineering

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Bangalore - 54

Evaluation Sheet

Sl. No	USN	Name	Content and Demon-stration (15)	Speak-ing Skills (2)	Team work (2)	Neat-ness and care (2)	Effect-iveness & Productivity (4)	Total Marks (25)
1	1MS14CS058	M Sneha						
2	1MS14CS148	Mipsa Patel						
3	1MS14CS134	Tilak S Naik						
4	1MS14CS136	Vibha Karanth						

Evaluated by

Name: Parkavi. A
Designation: Assistant Professor
Department: Computer Science & Engineering, RIT
Signature:

HOD, CSE

Contents

Abstract

Each student has a different set of skills and strengths that they excel in. Their level of understanding in different concepts taught to them varies. It is important to identify their level of understanding, and one such metric to do so is Blooms Taxonomy of Learning Domains. Based on a students performance in questions belonging to different Blooms levels, he is classified into one of them. Identification of the cognitive domain of a students learning can help in improving his skills from one of the lower levels to higher levels that rely more on complex and abstract mental ability.

1 Introduction

Blooms Taxonomy of Learning Domains identifies six levels within the cognitive domain, which are Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. These domains range from the simple recall or recognition of facts, as the lowest level called knowledge, through increasingly more complex and abstract mental levels, to the highest order which is classified as evaluation.

Given a set of questions that aim at evaluating the abilities of students in different learning domains, this project determines which level the students are best at. Two classifiers based on Naive Bayes and Support Vector Machines are used to classify the students marks into one of the levels, and the accuracy obtained using these models is compared.

2 Literature Survey

[1] Automated analysis of exam questions according to blooms taxonomy - Nazlia Omara, Syahidah Sufi Harisa, Rosilah Hassana, Haslina Arshada, Masura Rahmata, Noor Faridatul Ainun Zainala & Rozli Zulkiflib
This paper proposes an automated analysis of the exam questions to determine the appropriate category based on this taxonomy. This rule-based approach applies Natural Language Processing (NLP) techniques to identify important keywords and verbs, which may assist in the identification of the category of a question.

[2] Data Classification using Support Vector Machine - Durgesh K Srivastava, Lekha Bhambu
In this paper, a novel learning method, Support Vector Machine (SVM), is applied on different data (Diabetes data, Heart Data, Satellite Data and Shuttle data) which have two or multi class. SVM method does not suffer the limitations of data dimensionality and limited samples. It can be seen that the choice of kernel function and best value of parameters for particular kernel is critical for a given amount of data.

[?] An Effective Support Vector Machines (SVM) Performance Using Hierarchical Clustering - Mamoun Awad, Latifur Khan, and Farokh Bastani
This paper proposes a new approach for enhancing the training process of SVM when dealing with large data sets. It is based on the combination of SVM and clustering analysis. The idea is as follows: SVM computes the maximal margin separating data points; hence, only those patterns closest to the margin can affect the computations of that margin, while other points can be discarded without

affecting the final result.

[?] **Naive Bayes Models for Probability Estimation - Daniel Lowd, Pedro Domingos**
This paper proposes Naive Bayes models as an alternative to Bayesian networks for general probability estimation tasks. Experiments on a large number of datasets show that the two take similar time to learn and are similarly accurate, but Naive Bayes inference is orders of magnitude faster.

3 Algorithm

4 Implementation

The following 2 pages show an IPython Notebook that implements the algorithm.

```
In [1]: import pandas as pd
        import numpy as np
        from sklearn.naive_bayes import MultinomialNB
        from sklearn.svm import LinearSVC
        from sklearn.metrics import accuracy_score
        from sklearn.model_selection import train_test_split
        import matplotlib.pyplot as plt

In [2]: student_data = pd.read_csv('tst_student.csv', index_col=0)
        question_data = pd.read_csv('tst_questions.csv', index_col=0)

In [3]: student_data.head()

Out[3]:            1    2    3    4    5    6    7    8    9   10  Target
        Roll No
        1        2.5  1.0  5.0  4.0  2.0  5.0  2.5  3.0  4.0  3.5       4
        2        4.0  1.5  7.0  5.5  3.5  4.0  3.5  4.5  5.5  4.5       4
        3        3.5  1.5  5.5  6.5  5.0  5.5  4.5  3.0  5.5  5.5       1
        4        3.0  2.0  6.0  4.5  4.0  5.5  4.5  5.0  6.5  4.0       6
        5        3.5  2.0  6.5  7.0  4.5  5.5  5.0  3.5  5.5  4.5       3

In [4]: question_data

Out[4]:     Max Marks  Bloom Level
        Q#
        1           4            4
        2           2            2
        3           7            4
        4           7            2
        5           5            1
        6           6            1
        7           5            3
        8           5            2
        9           7            6
        10          7            5

In [5]: train, test = train_test_split(student_data, test_size=0.3)
        train_x, train_y = train[train.columns[:10]], train['Target']
        test_x, test_y = test[test.columns[:10]], test['Target']

In [6]: nb_model = MultinomialNB()
        svm_model = LinearSVC(multi_class='ovr')

In [7]: nb_model.fit(train_x, train_y)

Out[7]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

In [8]: svm_model.fit(train_x, train_y)
```

```
Out[8]: LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
             intercept_scaling=1, loss='squared_hinge', max_iter=1000,
             multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
             verbose=0)
```

```
In [9]: first_n = 20  # Number of data points to be used for plotting
```

```
In [10]: nb_prediction = nb_model.predict(test_x)
         svm_prediction = svm_model.predict(test_x)
         pd.DataFrame(data={'Expected': test_y.values, 'Naive Bayes': nb_prediction,
                           'SVM': svm_prediction}, index=test_x.index).head(first_n)
```

Out[10]:

Roll No	Expected	Naive Bayes	SVM
2965	6	6	6
5099	2	3	2
1213	3	3	3
1314	4	6	4
6443	5	5	5
4276	6	3	6
4446	4	3	4
9907	1	3	1
8514	5	3	5
3046	6	6	6
4771	1	3	1
3029	3	3	3
3521	4	3	4
6965	4	3	5
8721	3	3	3
6015	4	3	4
6634	5	5	5
9591	3	3	3
443	3	3	3
8957	6	6	6

```
In [11]: plt.plot(test_y.values[:first_n], 'gX')
         plt.plot(nb_prediction[:first_n], 'r')
         plt.plot(svm_prediction[:first_n], 'b--')
         plt.xticks(range(first_n), test_x.index[:first_n],
                    rotation=70, horizontalalignment='right')
         plt.show()
```

```
In [12]: accuracy_score(test_y.values, nb_prediction)
```

```
Out[12]: 0.46000000000000002
```

```
In [13]: accuracy_score(test_y.values, svm_prediction)
```

```
Out[13]: 0.88866666666666672
```

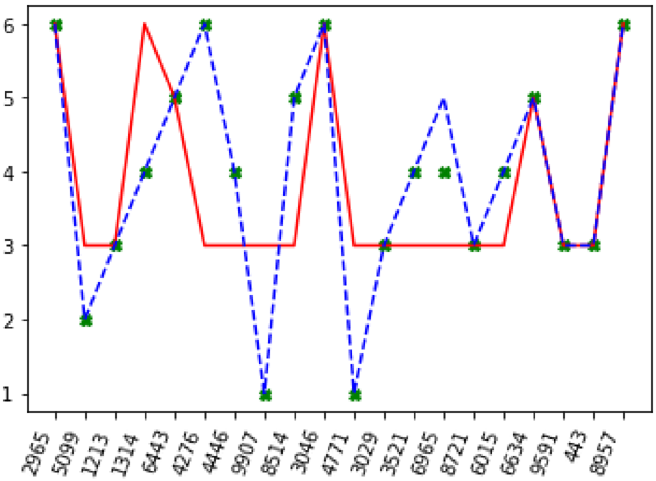

Figure 1: The plot of Student ID vs Bloom Knowledge is shown. The green crosses indicate the actual target, the red solid lines indicate the predictions by Naive Bayes algorithm, and the blue dashed lines indicate the predictions by SVM.

5 Results and Discussions

Naive Bayes classifier gives an accuracy of $\approx 46\%$ without normalization, and $\approx 30\%$ with normalization. We observe that the accuracy of the Naive Bayes model does not increase significantly even on increasing the size of the dataset used to train the model.

Comparatively, a model based on Support Vector Machines to obtain the Blooms level of a student gives an accuracy of $\approx 88\%$.

A comparison for prediction on 20 student's data is shown in Figure 1.

6 Conclusion

Classification problems can be solved using various different approaches. In this project, we explore the use of two such algorithms. Naive Bayes algorithm gives a low accuracy in identification of blooms level. If we normalize before learning, the accuracy drops further. Support Vector Machine gives us a fairly good accuracy.

References

[1] Nazlia Omar, Syahidah Sufi Haris, Rosilah Hassan, Haslina Arshad, Masura Rahmat, Noor Faridatul Ainun Zainal, and Rozli Zulkifli. Automated analysis of exam questions according to bloom's taxonomy. *Procedia - Social and Behavioral Sciences*, 59(Supplement C):297 – 303, 2012. Universiti Kebangsaan Malaysia Teaching and Learning Congress 2011, Volume I, December 17 20 2011, Pulau Pinang MALAYSIA.

[2] Durgesh K. Srivastava and Lekha Bhambhu. Data classification using support vector machine.

[3] Mamoun Awad, Latifur Khan, Farokh Bastani, and I-Ling Yen. An effective support vector machines (svms) performance using hierarchical clustering. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '04, pages 663–667, Washington, DC, USA, 2004. IEEE Computer Society.

[4] Daniel Lowd and Pedro Domingos. Naive bayes models for probability estimation. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 529–536, New York, NY, USA, 2005. ACM.