**Scribble – Confluence of NoSQL and Machine Learning**

**Introduction**

Scribble is a great example of a software which utilises NoSQL power of MongoDB with Python. It manifests the data via pymongo engine to predict the chances of admission based on various factors like GRE score, TOEFL score to name a few. It deploys the ML algorithm of Support Vector Machine tuned to its parameters to find the percentage chances of admission in the college. The project has employed the power of python interpreter to harness the utility of this project. The factors on which the decision is taken are GRE, TOEFL, Letters of Recommendation, SOPs, University Rating, whether they are research scholars or not, CGPA etc.

**Related Work**

1. Existing Works
2. This research[1] achieved ground-breaking results by solving problems related to Natural Language Processing by importing data from MongoDB.
3. MongoDB and python have also been used to carry out forensic research[2]. This helped in managing the results better given the advantage of schema independence in MongoDB.
4. Another work on MongoDB, involved creating a textbook management system that enables all the facilities a library possesses[3].
5. MongoDB has also found its utility in IoT and Big Data, by collecting the sensor data and RFID data.[4]
6. Benefits

Although many works have been published on this topic but most of them lack either proper management of data, effective prediction or very difficult interface to operate it. The project worked upon has covered all the loopholes and bottlenecks which were unseen before, ignored or unsolved. This project employs interface simplicity via Tkinter library and easy connectivity through PyMongo[5] Engine.

1. Drawbacks

Every bright side has a dark spot to it. Although tried but still the project lacked the attractiveness of the interface. It is the interface which starts the project, therefore it is very essential to work upon this. Since Tkinter[6] does not provide any support for such things. Other libraries available for the same but with much more support are pyglet[7], pythonqt[8], wxpython[9] etc.

**Implementation Methodology**

1. Algorithm

The algorithm used to predict the chances of admission is SVM[10] regressor. The support vector machine works on the principle of separation like-behaving points via a hyperplane. SVM considers each instance as a point in (n+1)-dimensional plane (n being the number of factors in consideration) and separates the points using a hyperplane. The hyperplane is selected such that the margin is maximum.

Other part and very important part of this project is the incorporation of MongoDB which is done via PyMongo engine. This library was developed by Mongo Inc. using this one can update, delete, secure and get a peek at the data. Main advantage of using MongoDB is that it supports sharding, which is deploying more than one server to better the exiting performance. Yet another ecstatic feature of MongoDB is that one does not need to define the schema prior to the work. It is always secondary to the work.

1. Programming Language

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python[11] has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. It has a very robust library based architecture which makes it very easy to use and debug. Not only researchers but many big conglomerate businesses use it to achieve the results they want.

1. Backend (NoSQL Support)

NoSQL support used to power the project is MongoDB which is a very versatile NoSQL database management system. It was developed by Mongo Inc. to cater the needs of the ever-growing amount of data i.e. Big Data. All the others systems present prior to the launch of MongoDB weren’t able to solve the problem of having the multiple types of data e.g. JSON, BSON etc. It is also preferred platform when the developer needs to connect multiple related documents and get cloud support.

**Project Contribution**

1. Idea

The project unleashes the combined power of NoSQl and Tkinter to develop a software that can predict the chances of students to get admitted in a college given some data. This would be helpful for the students to decide the steps they need to take to nurture their future. It uses SVM (Support Vector Machine) to predict the odds of admission using facts like GRE and TOEFL Score, University Rating etc. The software also enables the students to view relationship between various factors visually via graph. This mechanism is in-built in the software.

1. Motivation

Ourselves being college students very well identify the satisfaction of being assured. Other than hard work and dedication, a student also requires a tool to assure himself of his selection in her/his college. This utter need urged us to create an efficient and reliable tool which is omnipotent to bring you luck and assurance. Understanding the needs of our peers and mates is very important to create such a project.

**Conclusion**

The primary goal of this project is to provide a means to students to calculate their chances of admission. In order to do this, we have made an interface which employs the dual power of Tkinter and MongoDB to help students get the chances of admission. What students need to do is to only enter the data they have in the columns provided and check their scores. Not only this they can also visualise the existing data and can check out what factors affect their chances the most.

They can also retrieve the data corresponding to other data. This will provide means to students to check their friends’ data. From the data, what we could make out was that GRE and TOEFL score have a lot to do with their admission. University Rating, SOP are don’t matter as such but if it is better, it is an added advantage. Chances are highly dependent on CGPA and Research.

**References**

[1] Perkins, J. (2010). *Python text processing with NLTK 2.0 cookbook* (No. 1). Packt Publishing Ltd.

[2] Yoon, J., Jeong, D., Kang, C. H., & Lee, S. (2016). Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study. *Digital Investigation*, *17*, 53-65.

[3] Wei-Ping, Z., Ming-Xin, L. I., & Huan, C. (2011, May). Using MongoDB to implement textbook management system instead of MySQL. In *2011 IEEE 3rd International Conference on Communication Software and Networks* (pp. 303-305). IEEE.

[4] Kang, Y. S., Park, I. H., Rhee, J., & Lee, Y. H. (2016). MongoDB-based repository design for IoT-generated RFID/sensor big data. *IEEE Sensors Journal*, *16*(2), 485-497.

[5] Nielsen, F. Å. (2014). Python programming—databasing.

[6] Shipman, J. W. (2013). Tkinter 8.4 reference: a GUI for Python. *New Mexico Tech Computer Center*.

[7] De Smedt, T. (2013). *Modeling Creativity: Case Studies in Python*. University Press Antwerp.

[8] Summerfield, M. (2007). *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*. Pearson Education.

[9] Rappin, N., & Dunn, R. (2006). wxPython in Action.

[10] Scholkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[11] van Rossum, G., & de Boer, J. (1991). Interactively testing remote servers using the Python programming language. *CWi Quarterly*, *4*(4), 283-303.