Accelerating University Admission System using Machine Learning Techniques

Basil Alothman
Kuwait College of Science and Technology
Kuwait-City, Kuwait
b.alothman@kcst.edu.kw

Hanadi Alazmi
Kuwait College of Science and Technology
Kuwait-City, Kuwait
161152@student.kcst.edu.kw

Mohammad Bin Ali
Kuwait College of Science and Technology
Kuwait-City, Kuwait
161157@student.kcst.edu.kw

Noor Alqallaf
Kuwait College of Science and Technology
Kuwait-City, Kuwait
161129@student.kcst.edu.kw

Murad Khan
Kuwait College of Science and Technology
Kuwait-City, Kuwait
m.khan@kcst.edu.kw

***Abstract*—In this paper, we proposes a solution to enhance university admission system using different machine learning techniques like K-Nearest Neighbour(KNN), Decision Tree(DT) and Random Forest(RF). A deterministic algorithm is used to create a basis for the analysis, followed by several probabilistic algorithms, KNN, decision tree, and random forest. The outcomes are compared and a consensus is reached in categorising each student whether they got accepted, placed in foundation or rejected in the desired institute. Processing university applications is time-consuming for admission staff members, it is not efficient in terms of cost and time, thus it can sometimes be an overwhelming task, the chances of bribes occurring, or discrimination is highly likely to occur. The solution for this problem is to automate the application process thus using the help of a software with several embedded AI algorithm that improves the efficiency and reliability of the application process.**

***Index Terms*—University, Admission, Machine-Learning**

I. Introduction

The aim of this this paper is to develop an administrative software that makes the processes of university admission as easy and as efficient as possible for the administration staff, and as fair as possible to the students applying the desired institute. The software have an embedded machine learning algorithm that give an expectation for the student status based on a training data that is to the algorithm before the computations could be carried out. The software should have an integrated Graphical User Interface (GUI) that is simple and easy to use, thus the software can be run by a person with limited knowledge about computers and Information Technology (IT). Although the software is written in python[1], it is then converted into an executable self-installing package. The user is not responsible for the backend of the software and all interactions and changes could be done from the front end. However, it was considered that some changes could occur to the acceptance criteria such that the condition script was left to be edited using a simple text editor. The software also compared deterministic results with the statistical results obtained. Similarly, if the results show contradiction is terms of the student's status, an interview will be booked with the student. The interview is carried out by the administration team and will then determine the status of the student based on the results of the interview. Nevertheless, the interview could be replaced with an entrance exam or any sort of qualification validation to ensure that the student deserves his place and there are no wasted resources or talents.

This paper presents the Grade Point Average (GPA) rank score for the University admission that has developed by the handle of student's school grade along to increment admission standards such as segregated minorities and high-performance individuals. A "GPA-ranking score" popularized in 2012 by Chile's university is different from other rankings. The Chilean version is comparable to the combined bonus to the GPA without replacing the national admission exam, and it motivates both academic effort and GPA inflation. They use to change and replicate instruments methods by using the Chilean ranking. The Chilean goal is to rebuild GPA and achievement to analyses the Chilean reforms on GPA ranking and achievements based on student distributions and larger schools, finally the improvement of the university admission rate probabilities. The Chilean experience has some similarities with our local university admission, like GPA admission grades, and they have students from private and public schools. Also, they have private and public Universities, like our experience here in Kuwait [2]. This paper suggests that schools sometimes contribute in GPA inflation, giving students a higher GPA than they deserve to maintain their reputation which is not a sustainable practice on the long term. This is often the result of steak holders pressuring school administrators. To overcome this limitation, other factors will be introduced to the algorithm to achieve more accurate and reliable results. The rest of the paper is divided into the following parts. Section II explains the recent literature published in the domain of

[1]Programming Language

the proposed research work. Section III gives the detail of the proposed scheme. Analysis and results are discussed in Section IV. Finally, the conclusion is given in Section V.

II. Related Works

Robu, A., Filip, and Robu, R. conducted a detailed assessment of the benefits generated by using online learning platforms in institutions of higher learning. They assert that e-learning platforms create a favorable framework for the students to optimize their education in preparation for examinations. Students are offered both a more comfortable learning landscape and an opportunity for self-evaluation. By analyzing their progress, learners can project their level of preparation at different moments in time. Additionally, this study employs the use of visual elements as its primary advantage [3].

In a separate study, researchers Handa and Gordon investigated the impact of university admission policy on the performance of part-time and full-time students in developing countries. Their article reported that the success rate of full-time students is almost twice that of part-time students at the University of West Indies. The scholars attributed this disparity to the differences in the choice of major, demographic properties, and pre-entry qualifications. Their use of primary data and direct comparisons enhances the credibility and accuracy of their study although it was not based on machine learning [4]. Over recent years, various researchers have conducted studies to investigate the significance of automated admissions systems. Mengash proposed a framework that higher education institutions can capitalize on in making more informed decisions in admission processes. She asserted that tertiary learning institutions should make predictions on learners' academic performance before admitting them. The study constituted the provision of models, which, if utilized appropriately, can optimize higher learning institutions' limited resource distributions. These models are Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Artificial Neural Network (ANN) [10]–[12]. In the findings of Mengash [5], it appears that ANN yielded the highest accuracy of 79% making it superior against other classification methods proposed in her research.

The University of Texas at Austin Department of Computer Science (UTCS) has been advanced with a statistical machine learning system to support the work of the graduate admission. GRADE developed how to anticipate each new applicant by using historical admission data. Also, they implement for the reviewers to spend most of their time and focusing their attention on parts of each applicant's file [6]. The proposed software has LR encoded which achieved an accuracy of 84%. The researchers also experimented with SVM; however, the model was difficult to interpret and Logistic regression was chosen [6].

Boddy and Dean described the problem-solving tasks called Deliberation scheduling, empowering designing systems that are efficient in taking their own calculations source into the application during the planning and problem-solving. The design of systems that dominate their calculation source by using the assumption of decision-making procedures and choice over the outcomes result from applying those actions. Deliberation scheduling affects the accurate allotment of calculations source to decision-making procedures based on hoping of those actions on the system's performance [7].

Braun, Dwenger, Kubler, and Westcamp [8] supported the idea of quota effectivity in maximizing match outcomes among students and college programs and courses. Employing two mechanisms, mainly sequential and simultaneous, the researchers were able to conclude how the is proven more effective in reserving the number of reserved seats in a quota course to more deserving students. Furthermore, employing two mechanisms may yield more favorable and reliable results. The machine intelligence that goes into sequentially and simultaneously predicting the fittest programs for applicants and produces comparable results that, when assessed, present relationships within the results yielded by the system [8]. Consequently, for students admitted in their first choices, there appears to be a higher probability of more favorable educational and economic outcomes [9]. Specifically, in terms of academic prosperity, there appears to be a better chance for an individual to accomplish a Master's degree if he has been admitted to his first-choice. This seems to implicitly state that students who did better in high school and are more competent in terms of academic performance are more secured of a promising future that lies ahead of them in terms of employment and career opportunities. For their less-educated counterparts, it may be challenging to maximize opportunities in employment because of higher standards set to be considered one of the bests.

Large data sets can be analyzed to predict future outcomes of certain events through advanced statistical analysis [13]–[15]. The prediction however can be done by humans but is time consuming and the model might be susceptible to high uncertainties and randomness due to human error since the task can get quite complex. To overcome this, the prediction method could be automated by feeding the training data to an algorithm that can then predict the outcome of a new dataset based on the training data. Some machine learning algorithms and techniques were introduced by Obulesu et al [16], such as decision tree, KNN, SVM and NB.

III. Proposed Scheme

The proposed design is a fully visual and interactive software application that conduct all calculation and analysis though several algorithms that are hidden away from the user in the backend of the application. The software is developed fully in python using anaconda as the package manager, it is also important to note that the final product does not require python to be installed on the machine. The software is compiled such as it is self-installed on the PC in a way that anyone can use the software to its full potential. The app developed has the capability of processing thousands of data entries without any issues at a fast pace. In addition, the software classifies the students as Accepted, Foundation, or

Rejected based on the Kuwaiti Universities admission requirements, these standards and requirements are obtained from several sources and they all relate to the equivalent percentage. The results from the software are exported in a CSV format that can be processed further by the administration team. In terms of the algorithms used, as discussed earlier the software follows two classes of algorithms; deterministic algorithm and probabilistic algorithm. The deterministic algorithm is the core of the software since the output of the deterministic portion of the software is used as one of the inputs into the probabilistic portion of the software. In Simpler terms the output of the algorithm that determines the status using a set of "if, if else and else" functions is used as the testing data in conjunction with the training data imported by the user that corresponds to the desired institute, the software will then check the choices the user has made in the GUI on what machine learning algorithms were chosen which will adjust the output accordingly. There are three machine learning algorithms embedded, K-nearest neighbour, decision tree and random forest. Initially there were supposed to be only two machine learning algorithms, however it was found out at later stages of development that the decision tree algorithm for the data being dealt with suffers from extreme overfitting. To overcome this problem the random forest algorithm was embedded which is basically and improved version of the decision tree algorithm.

Overview of the proposed system High School Percentage (HSP), Equivalent Percentage (EQP), Majors: Medicine (Med), Mathematics (M), Pharmacy (Ph), General Health (GH), Architecture (Arc), Engineering (Eng), Life Science (LS), Administrative Sciences (AS), Education Scientific (ES), Education Literary (EL), Arabic Literature(AL), English Literature (EL), Shariaa (Sh), Courses needed: English(E), Arabic(A), Math(M), chemistry(C).

$$EQP_{Med} = 0.15 \times E + 0.1 \times M + 0.1 \times C + 0.65 \times HSP \quad (1)$$

$$EQP_{Eng} = 0.15 \times E + 0.20 \times M + 0.65 \times HSP \quad (2)$$

$$EQP_{Ph} = 0.15 \times E + 0.1 \times M + 0.1 \times C + 0.65 \times HSP \quad (3)$$

$$EQP_{GH} = 0.15 \times E + 0.15 \times M + 0.70 \times HSP \quad (4)$$

$$EQP_{Arc} = 0.15 \times E + 0.15 \times M + 0.70 \times HSP \quad (5)$$

$$EQP_{LS} = 0.15 \times E + 0.15 \times M + 0.70 \times HSP \quad (6)$$

$$EQP_{AD} = 0.15 \times E + 0.15 \times M + 0.70 \times HSP \quad (7)$$

$$EQP_{ES} = 0.075 \times E + 0.075 \times M + 0.05 \times A + 0.80 \times HSP \quad (8)$$

$$EQP_{EL} = 0.1 \times E + 0.1 \times A + 0.80 \times HSP \quad (9)$$

$$EQP_{AL} = 0.15 \times A + 0.85 \times HSP \quad (10)$$

$$EQP_{EL} = 0.15 \times E + 0.85 \times HSP \quad (11)$$

$$EQP_{SH} = 0.15 \times A + 0.85 \times HSP \quad (12)$$

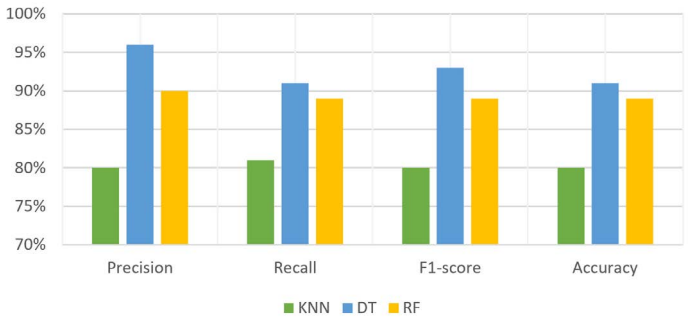

Fig. 1. Metrics for the Admission to Colleges of Engineering

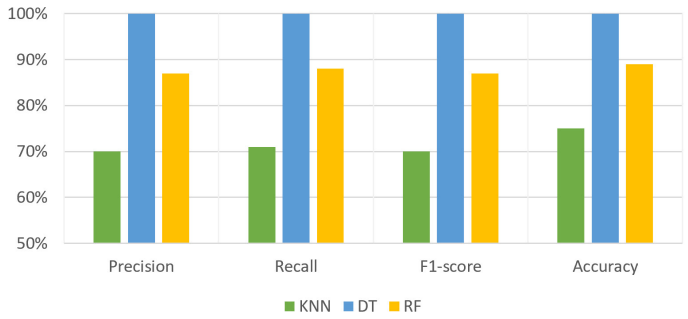

Fig. 2. Metrics for the Admission to Colleges of Arabic Literature

Results and Analysis During the testing phase, the software was allowed to process numerous data sets with all the options toggled on to check for any bugs or possible crashes. Initially there were a lot of issues with the code as the software kept crashing. These issues were fixed gradually, eventually the software was optimised to run smoothly thus enhancing the user experience.

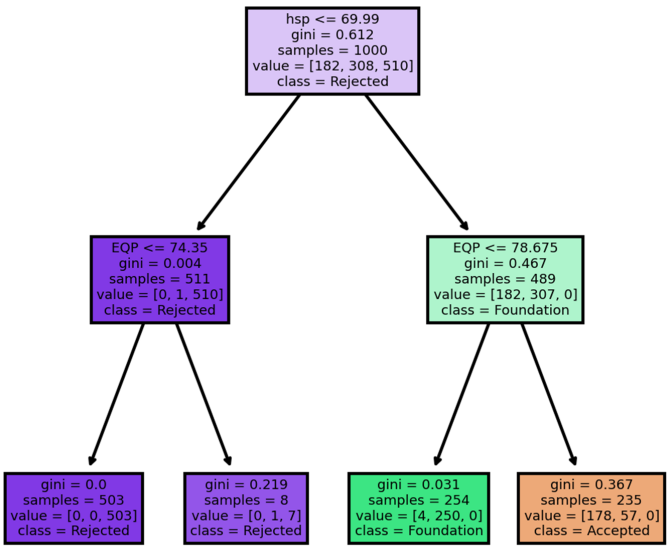

Fig. 3. Illustrates the visual output of the decision tree

TABLE I
SHOWS THE OUTPUT FOR THE COMBINED OPTION

ID	M	E	C	A	F	HSP	EQP	ST	KNN	DT	RF	Int
1	84.46	76.12	66.17	62.2	46.21	66.36	71.1	RJ	RJ	RJ	RJ	No
2	56.05	52.87	40.09	90.86	59.24	58.1	56.7	RJ	RJ	RJ	FD	Yes
3	54.65	80.35	53.17	56.49	83.79	67.83	66.35	RJ	RJ	RJ	RJ	No
4	49.92	59.43	63.29	66.27	61.95	57.19	55.7	RJ	RJ	RJ	RJ	No
5	76.43	78.05	83.46	59.08	70.61	68.61	71.1	RJ	FD	RJ	RJ	Yes
6	76.9	61.33	86.03	47.53	55.53	66.61	67.25	RJ	RJ	RJ	RJ	No
7	95.65	49.15	54.44	78.4	74.62	74.56	74.45	FD	FD	FD	FD	No
8	97.49	45.07	42.13	81.74	68.05	64.08	67.75	RJ	RJ	RJ	RJ	No
9	88.11	84.33	97.14	95.8	83.86	82.79	83.5	AT	FD	AT	AT	Yes
10	76.76	90.24	71.83	86.77	46.3	68.79	72.9	RJ	FD	RJ	RJ	Yes

TABLE II
SHOWS THE OUTPUT FOR THE COMBINED OPTION

ID	M	E	C	A	F	HSP	EQP	ST	AP	Int
1	84.46	76.12	66.17	62.2	46.21	66.36	71.1	RJ	RJ	No
2	56.05	52.87	40.09	90.86	59.24	58.1	56.7	RJ	RJ	No
3	54.65	80.35	53.17	56.49	83.79	67.83	66.35	RJ	RJ	No
4	49.92	59.43	63.29	66.27	61.95	57.19	55.7	RJ	RJ	No
5	76.43	78.05	83.46	59.08	70.61	68.61	71.1	RJ	FD	Yes
6	76.9	61.33	86.03	47.53	55.53	66.61	67.25	RJ	RJ	No
7	95.65	49.15	54.44	78.4	74.62	74.56	74.45	FD	FD	No
8	97.49	45.07	42.13	81.74	68.05	64.08	67.75	RJ	RJ	No
9	88.11	84.33	97.14	95.8	83.86	82.79	83.5	AT	FD	Yes
10	76.76	90.24	71.83	86.77	46.3	68.79	72.9	RJ	FD	Yes

The deterministic model to calculate the equivalent percentage was tested against has calculations and the results matched for all the institute. However, a mistake was found in the department of English literature where the equivalent percentage was calculated incorrectly inside python. All other results from python matched the handwritten calculations. The probabilistic model on the other hand, hence the machine learning was tested using a confusion matrix, Precision, Recall, F1-score, and accuracy. The confusion matrix gives a general idea about the performance of the machine learning model in terms what predictions were accurate and what predictions were not, this was done by comparing the testing data with the training data that is imported by the user. Figure 1 and Figure 2 illustrates the output in terms of the metrics for each institute that the software processes. Each figure demonstrates the precision, F1-score, Recall, and accuracy for each machine learning algorithm used. From the results obtained in terms of the machine learning algorithm, it was evident that the decision tree algorithm had the highest accuracy for all the institutes. While the KNN algorithm showed the lowest accuracy. The random forest classifier on the other hand was in the mid-range in terms of accuracy. Nevertheless the decision tree showed signs of overfitting since most of the times the metrics were maxed at 100%. This can be related to the fact that the database used for training and testing were both randomly generated. However further testing is required to validate the results and the reliability of the software. The main challenge that was faced was to properly link and affiliate the Graphical User interface with the logic of the algorithm, since most of the bugs and errors that were faced were in regard to this issue. Interview (Int), Status (ST), Foundation (FD), Rejected (RJ), Accepted (AT), Admission Prediction (AP).

Regarding the exported output, the results will mainly depend on the input of the user. Table 1 shows a sample output if the user chooses the machine learning options to export individually. The export file will show the student details such as the ID, all the skills test results, and the calculated equivalent percentage for the specified institute. The 9th row shows the output of the deterministic algorithm, which the 10th column shows the output for the probabilistic algorithm. The 11th row compare the value from the 10th row and the 9th row, if the values are different the student will be assigned an interview or any sort of qualification check. If the values were equal to each other the status of the student is set. Figure (26) shows the output for one machine learning algorithm On the other hand, if the user chooses the combined output from the user interface, the output is similar to Table 2. Similar to the previous figure shown, the first row shows the unique ID, then all of the students' grades will be listed along with the Equivalent percentage calculated for the desired institute. The 9th column is the result from the deterministic model, while columns 10, 11, and 12 are the output of the probabilistic model. If all values in the output columns are the same, the student status will remain unchanged, while if 1 column is different than the other 3, the student will be assigned an interview or any form of qualification check.

The confusion matrix is another form of testing/validating the results. The software has an option to be run in console mode where the metrics shown above, and the confusion matrix will be automatically generated and displayed. The confusion matrix is a strong tool that help visualise the results in a form of a matrix. Table (2) illustrates how these results could be interpreted.

TABLE III
ILLUSTRATES THE CONFUSION MATRIX

Student Admission Status	**Machine learning Level**		
	Accepted	***Foundation***	***Rejected***
Accepted	X1 - True	X2 - False	X3 - False
Foundation	X4 - False	X5 - True	X6 - False
Rejected	X7 - False	X8 - False	X6 - True

IV. Conclusion and Future work

To conclude, a solution was proposed to enhance the processes of admissions to university. A software was developed that processes data using several algorithms that can be split into two classes, deterministic algorithm, and probabilistic algorithms. The deterministic algorithm is based on a series of if functions that will check the equivalent percentage of the student against the requirements to be accepted in the desired institute. The probabilistic algorithms on the other hand are based of machine learning techniques that will consider more attributes when determining the status of the student. In total three algorithms were used, KNN, decision tree, and the Random forest classifier. The algorithms were optimised to suit the training data, however since the data was randomly generated, the algorithm suffered from overfitting.

Additionally, a visual output for the decision tree was also included in the algorithm, the output will illustrate the process

in which the machine learning algorithm will categorise the student based on the criteria. For the decision tree, the two attributes to be tested were the high school percentage (HSP) and the equivalent percentage (EQP). Figure 3 illustrates the visual output that is automatically generated and saved in the export environment that the user chooses. Figure 3 on the other hand illustrates the visual process of the random forest classifier which takes more steps to classify the entry when compared to the decision tree algorithm.

REFERENCES

[1] H. A. Mengash, "Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems," in IEEE Access, vol. 8, pp. 55462-55470, 2020, doi: 10.1109/ACCESS.2020.2981905.

[2] Fajnzylber, Eduardo Lara E., Bernardo León, Tomás. (2019). Increased Learning or GPA Inflation? Evidence from GPA-Based University Admission in Chile. Economics of Education Review. 72. 147-165. 10.1016/j.econedurev.2019.05.009.

[3] Robu, A., Filip, I. Robu, R. (2018). Online platform for university admission. IEEE.

[4] Handa, S. Gordon, P. (1999). University admissions policy in a developing country: evidence from the University of the West Indies. Economics of Education Review, 18 (1999) 279–289.

[5] Mengash, H.A. (2011). Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems. IEEE, Vol 8, 55462-55470.

[6] GRADE: Machine Learning Support for Graduate Admissions Austin Waters and Risto Miikkulainen Department of Computer Science 1 University Station 0500, University of Texas, Austin, TX 78712

[7] Boddy, M. and Dean, T.L. (1994). Deliberation scheduling for problem-solving in time-constrained environments. Artificial Intelligence, 67(2): 245-285

[8] Braun, S., Dwenger, N., Kubler, D., Westcamp, A. (2014). Implementing quotas in university admissions. http://dx.doi.org/10.1016/j.geb.2014.02.004

[9] Heinesen, E. (2018). Admission to higher education programs and student educational outcomes and earnings – evidence from Denmark. Economics of Education Review 63: 1-19. doi: 10.1016/j.econedurev.2018.01.002.

[10] Khan, M., Saad, M. M., Tariq, M. A., Seo, J., Kim, D. (2020, December). Human Activity Prediction-aware Sensor Cycling in Smart Home Networks. In 2020 IEEE Globecom Workshops (GC Wkshps (pp. 1-6). IEEE.

[11] Khan, M., Jan, B., Farman, H. (2019). Deep learning: convergence to big data analytics (p. 93). Singapore: Springer.

[12] Jan, Bilal, et al. "Deep learning in big data analytics: a comparative study." Computers Electrical Engineering 75 (2019): 275-287.

[13] Silva, B. N., Khan, M., Seo, J., Muhammad, D., Yoon, Y., Han, J., Han, K. (2018, December). Exploiting big data analytics for urban planning and smart city performance improvement. In 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS) (pp. 1-4). IEEE.

[14] Silva, B. N., Khan, M., Han, K. (2018). Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. IETE Technical review, 35(2), 205-220.

[15] Nathali Silva, B., Khan, M., Han, K. (2017). Big data analytics embedded smart city architecture for performance enhancement through real-time data processing and decision-making. Wireless Communications and Mobile Computing, 2017.

[16] O. Obulesu, M. Mahendra and M. ThrilokReddy, "Machine Learning Techniques and Tools: A Survey," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, 2018, pp. 605-611, doi: 10.1109/ICIRCA.2018.8597302.