Prediction of Graduate Admission using Multiple Supervised Machine Learning Models

Zain Bitar
Electrical Engineering Department
Princess Sumaya University for Technology
Amman, Jordan
zainbitarr@gmail.com

Amjed Al-Mousa
Computer Engineering Department
Princess Sumaya University for Technology
Amman, Jordan
a.almousa@psut.edu.jo

***Abstract*–In response to the highly competitive job market at present times, an increased interest in graduate studies has arisen. This has not only burdened applicants but also led to an increased workload on admission faculty members of universities. Any chance of abridging the admission process impelled applicants and faculty workers to look for faster, efficient, and more accurate methods for predicting admissions. The goal approach of this paper is to implement and compare several supervised predictive analysis methods on a labeled dataset based on real applications from the prestigious university of UCLA; Regression, classification, and Ensemble methods are all the supervised methods that are to be employed for prediction. The dataset relies profoundly on the academic performance of the applicants during their undergrad years. The coefficient of determination, as well as precision and accuracy, are the measures used to compare the different models. All predictive methods proved to show accurate results, however; certain methods proved to be more promising than others were. Predictions were obtained within short time frames, which in turn will cut down the time in the admission process.**

Keywords- admissions, graduate studies Regression, Classification, Ensemble methods, coefficient of determination

I. INTRODUCTION

Graduate programs have gained increased popularity seeing that most students find themselves looking for a chance to continue their education after completing their undergraduate study; this is usually a result of students gaining a better perspective of what educational field they would like to pursue. Nevertheless, most prestigious schools require a minimum standard of academic performance based on previously taken standardized exam scores and cumulative GPA as well as several other academic measures; This can be highlighted in several papers [1,2]. With this increased demand for graduate admission, registration offices have been pressured with overlooking thousands of applications.

The predictor of design will provide registration offices with a good overlook of hundreds even thousands of applications at a time with high accuracies; this is also very useful from an applicant's point of view where it saves them money on application fees and time such that the applicant can still get a chance of early admission. Highly correlated features with the admission rate will be highlighted; elucidating to the applicants what will affect their chances of admission. The methods will be trained to predict admission per the UCLA admission rules, which rely on the applicant's previous educational record. The predictor will not consider any non-educational related features.

Several Machine learning supervised learning algorithms will be utilized in this paper to predict the rate of acceptance as a percentage: SVM (support vector machines), Logistic Regression, Linear Regression, Decision Trees, and Random Forest. Regression models will be compared according to their coefficient of determination denoted by R^2 whilst classification models will be compared according to their accuracy, precision, and recall. Ensemble methods such as Boosting and bagging as well as stacking will be used to enhance the accuracy for the classifier; several studies have been focusing on the impact of Ensemble methods in the future of machine learning [3].

II. RELATED WORK

With the flourishing of machine learning in the twentieth century and its ability to make everyday tasks effortless, its applicability also extended in educational fields. A very interesting domain where machine learning was implemented within the educational field is the ability to predict the admission of an applicant into an educational institution. In several countries such as China, the application process could be competitive especially with the ability of students to get early admission. The approach was to employ supervised methods such as Logistic Regression on a university dataset to compute the prediction of students with different educational abilities on getting an early acceptance in admission [1]. Another take on machine learning applicability in the educational admission field is implementing a recommender that uses KNN to recommend the best-suited university for each applicant based on an academic dataset with features that test the academic performance identical to those in the dataset used [2].

Usually, datasets do not come in a single language especially as in [4] if you are building a universal predictor the data will more likely have unnecessary features. In [3] the predictor was able to use Decision Tree modeling along with recommender systems as in [2] are two strong modeling algorithms used in the prediction process. The predictor also provided applicants a view of what features have the highest effect in influencing their admission rate. Similarly, the predictor to be designed should also be able to use Decision Trees to provide good approximations; however, further enhancement of this model will be presented later in this paper.

The First take on this implementation is in [5] where it introduces a predictor known as GRADE; GRADE

978-1-7281-6861-6/20/$31.00 ©2020 IEEE

preprocesses numerical and categorical data such as GRE score, GPA and school name. The predictive analysis method of [4] is logistic regression infused with the log of odds; alike this paper performance measures such as the ROC and AUC will be computed, compared and enhanced as that is the approach of this paper.

III. Data Analysis& Preprocessing

The data of use [6] composed of five-hundred instances with no null value entries nor any categorical attributes; each instance in the dataset represented an applicant. This dataset has been acquired from UCLA's admittance history data. The number of attributes given in dataset is eight where all attributes are numeric:

I. GRE Score (General Record Examinations); this score measures general knowledge in undergrad Math and English. This score ranges from a value of 260 to 340

II. TOEFL Score (Test of English as a Foreign Language); this score measures students English abilities. This score ranges from a value 0 to 120.

III. SOP (Statement of Purpose); a letter written by the applicant explaining their purpose of the application. This is scored on a range from one to five.

IV. LOR (Letter of Recommendation); tests the weight of the recommendation provided by the applicant. This is scored on a range from one to five.

V. CGPA (Cumulative GPA); based on the academic performance of the applicant in undergraduate studies. This is scored on a range from one to ten.

VI. University Rating; based on the reputation of the applicant's previous university. This is scored on a range from one to five.

VII. Research Experience; binary value based on whether the applicant has any research familiarity. This value is either one or zero.

VIII. Chance of Admission; the rate of admission into graduate school. This attribute is the targeted value in which will be predicted as the rate from zero to one.

The dataset included a "Serial No." attribute, which had no benefit to the data frame considering it had the same value as the numeric index of the instances. To understand the distribution of the data and to gain a better statistical understanding Table.1 describes the data using Pandas (Python Data Analysis Library) describe function.

TABLE 1: The statistical characteristics of the Data

Label	Count	Mean	Std	Max	Min
GRE	500	316.4	11.29	340	290
TOEFL	500	107.2	6.08	120	92.0
Uni Rating	500	3.11	1.14	5.0	1.0
SOP	500	3.37	0.99	5.0	1.0
LOR	500	3.48	0.92	5.0	1.0
CGPA	500	8.57	0.60	9.0	6.8
Research	500	0.56	0.50	1.0	0.0
Admittance	500	0.72	0.14	0.97	0.34

As seen above the data scales differ profoundly across attributes and this can affect the SVM in fitting the data with the "widest fitting street". To avoid this effect a preprocessing technique known as Standard Scaling is imported from Scikit-learn; this technique was chosen over min-max scaling given that the data might include several outliers.

Correlation matrices help provide an intuitive idea about what predictive analysis methods would be more useful in finding a suitable model. Fig 1 employs Matplotlib and Seaborn libraries to plot the correlation matrix using color maps to help emphasize the number scale in colors.

It can be seen that all features have a positive uniform correlation with the admission rate. The highest impacting attribute on the admission rate is the "CGPA" at a substantial value of 0.88 whilst, "Research" has the lowest value between all features at a value of 0.54.

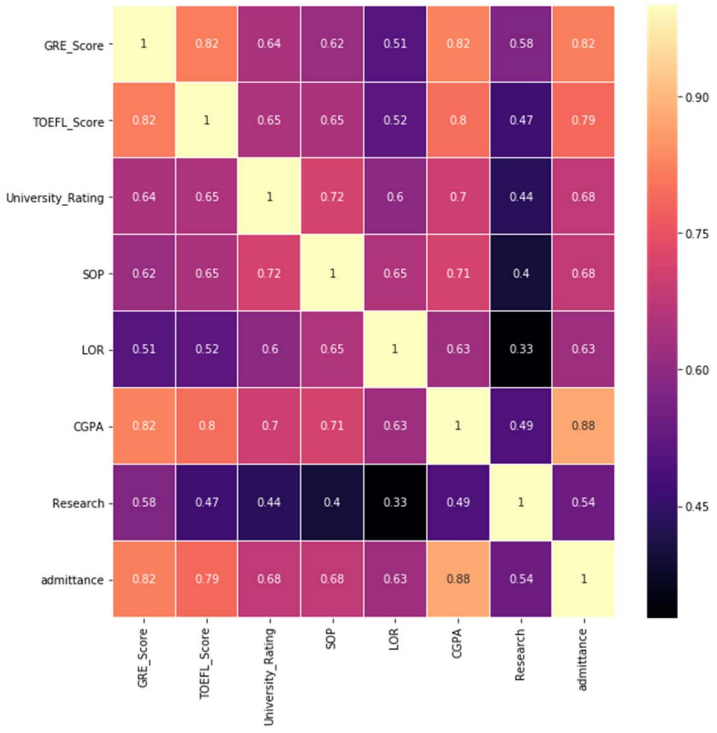

Fig.1: Correlation matrix using all eight attributes

Now to visualize the effect of the features within themselves, a scatter plot is shown in Fig.2 using Matplotlib scatter plot function. This plot emphasizes the relationships between the three highest correlated features with the admittance rate: "GRE_score", "TOEFL_Score", and "CGPA". As suggest by Fig.1, the "GRE_Score" along with "TOEFL_Score" have a high correlation among each other along with the "CGPA". To clarify this correlation a scatter plot Fig.2 was implemented to represent a 3D plot in a 2D plot using the color map feature. The scatter plot captures the linearly intrinsic trend amongst the three largest impacting features on the admittance rate.

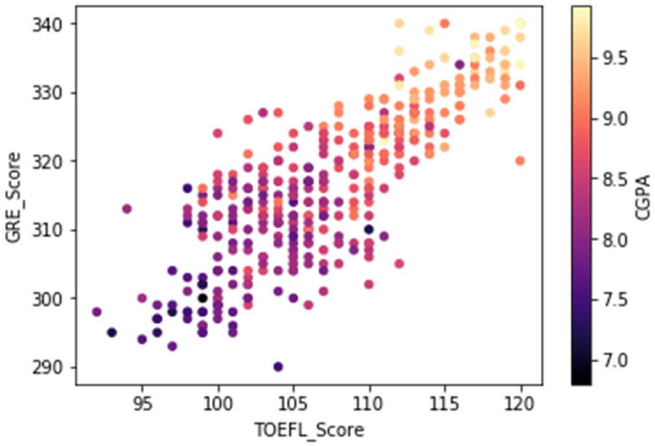

Fig.2: Correlation between the GRE_Score, CGPA, and TOEFL_Score

"University rating" which does not show a very promising correlation value in terms of admittance rate, will prove to have a clear correlation. This can be better elucidated using a bar graph under the assumption that an "admittance rate" above 0.75 guarantees acceptance of the applicant. Fig.3 shows how "University rating" actually does affect the "admittance rate" proportionally with a mild exception that the number of admitted applicants from a "University rating" of four is slightly greater than that of a five.

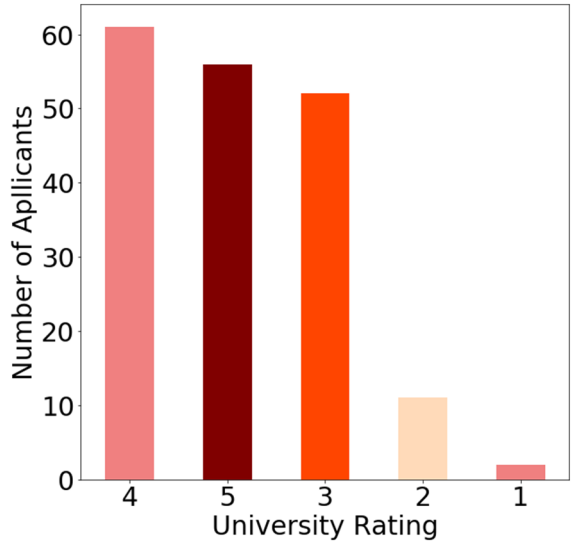

Fig.3: Number of applicants admitted along with their university rating

IV. Modeling Methods

Several supervised predictive analysis methods can be implemented on the data whether these methods categorize under Regression, Classification, and Ensemble (applicable to both regression and classification). Innately modeling the data set was more likely to be done using Regression algorithms; nevertheless, Classification can also be done by slightly altering the target attribute. An admittance rate value above 0.75 is considered as a definite admission or a binary 1 anything below will be considered as a binary zero: given that in Table 1 the mean of the admit rate is about 0.72 a value slightly higher will be a reasonable rate.

Ensemble techniques such as vote classifier, bagging, boosting and stacking algorithms will be introduced to improve the accuracy of the classification process; Ensemble methods usage has been widespread, as shown in papers [7,8] comparing and upgrading their performance in terms of accuracy. The above algorithms are used post splitting the data into test data (20%) and training data (80%).

A. Regression Models

Several different regression algorithms will be used to pursue the best model according to the performance measure R^2 (coefficient of determination). Based on Fig.1 and Fig.2 Linear Regression models seem promising. After importing the Linear Regression library from Scikit-Learn, the model was fitted and later used to predict the test data. To make sure the model was not overfitting the data, cross-validation was done using three folds and the scoring was based on the root mean square error found in Table 2. Three folds produced a decent variation of data and data size in each fold. Testing the performance of this model was done by computing the R^2score using the test data along with the predicted values.

SVR (Support vector machine regression) was used in expectation that the performance measure R^2 might increase; however, in contrast to linear regression SVR has several hyper-parameters that can be adjusted to realize the lowest mean square error: **ε** (epsilon) and **C**. Subsequently, to maximize the performance of SVM, GridSearch was imported from Scikit-Learn to establish the best hyper-parameter values of **C**=1.5 and **ε**=0.1 along with a linear kernel. Now cross-validation can be implemented with values found in Table 2. Now to test the performance of the model the same R^2score was evaluated on the test data along with the SVR predicted values

Now for the final regression algorithm, Decision Tree Regressors along with their ensemble Random Forest Regressors will be implemented. Similar to SVR Decision Tree Regressor hyper-parameters were computed using GridSearch, the same procedure will be repeated in this case of Decision Tree Regressors. These are the values acquired max depth=5 and min samples split=50. Cross-Validation scores of both algorithms are found in Table 2. The R^2score was also evaluated on the predicted values of both algorithms.

TABLE 2: Cross-validation scores using all four-regression algorithms

Regression	First Fold	Second Fold	Third Fold
Linear Regression	0.0643	0.0615	0.0568
SVR	0.0668	0.0724	0.0675
Decision Tree Regression	0.0765	0.0737	0.0667
Random Forest Regression	0.0702	0.0654	0.06190

A bar graph using all four algorithms R^2 was used to compare which algorithm had the best performance in terms of modeling the data in Fig.4.It can be deduced that Linear Regression is the best-fit algorithm and it matches the data with a R^2score of 0.819. The second best-fitting model is the Random Forest Regressors at a value 0.792: Ensemble algorithms are proven to improve performance especially in the case of weak classifiers.

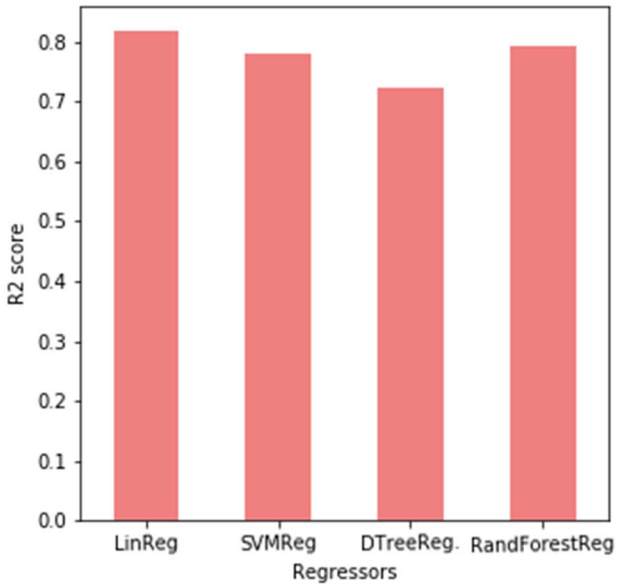

Fig.4: R^2score of four different regression algorithms.

B. Classification Models

After splitting the data into training and testing sets, the admittance rate also known as the target feature is transformed into a binary value, where if the admittance rate is greater than 0.75 is considered as a binary one anything else is a zero. Three different classification algorithms models Decision Trees, Logistic regression, and SVC (support vector classifier) are to be compared based on accuracy, recall, precision, and AUC.

The first algorithm to be used is Logistic Regression; this algorithm relies on a sigmoid function and a threshold. Similar to the Linear Regression algorithm the Logistic Regression places weights on each feature with a cost function (1). The algorithm works on minimizing the cost function using gradient descent.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\, log\left(p^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - p^{(i)}\right)\right] \qquad (1)$$

GridSearch was used to find the optimal generalization factor "**C**" value, which was found to be one. This model provided an adequate accuracy as shown in Table 3. To ensure the model did not over-fit the data cross-validation was done; moreover, the Cross-validation score of this classifier based on the accuracy using three folds has a mean value of 0.8946. Precision and recall values can be readily calculated from the confusion matrix in Fig 5.

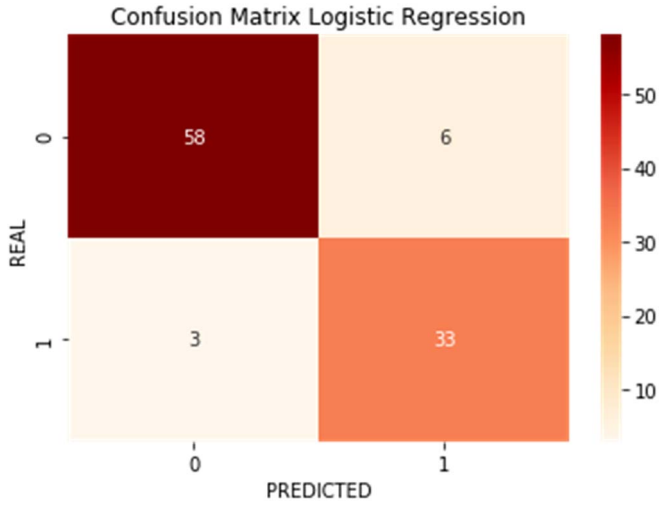

Fig.5: Confusion matrix of the predicted values using Logistic Regression.

The second algorithm used is SVC; this classifier is known as a strong classifier, with minimum computational power. All hyper-parameters are interpreted using GridSearch with the following results C=10, degree=1.0, gamma= 0.01, and kernel ='linear'. The mean cross-validation score of three folds is 0.89663; given such a high cross-validation score the model looks promising. A confusion matrix, shown in Fig.6, is used to show the model's performance measures.

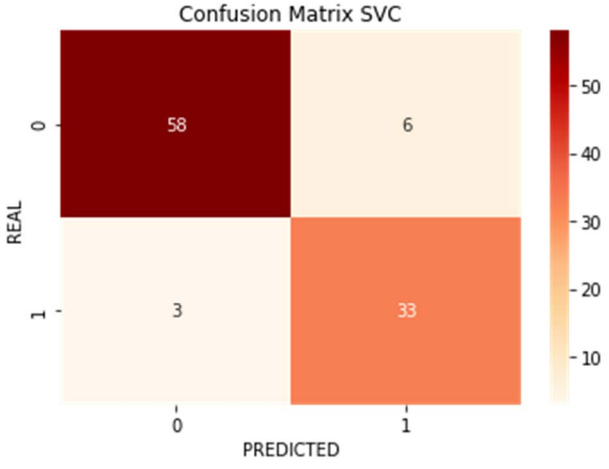

Fig.6: Confusion matrix of the predicted values using SVC.

The final classification algorithm to use is the Decision Tree algorithm. Even though this algorithm is very trivial and considered as a weak learner, it has a rather logical approach. The cost function in this algorithm is based on the CART principle. CART is known for being a greedy algorithm: it looks for the best way to find a split at the current layer without taking into account the best split for further layers. Equation (2) represents the CART cost function and its reliance on the gini index G and the number of instances m; G_{left} and G_{right} represent the gini measure, while m_{left} and m_{right} represent the total number of instances to the left and right respectively.

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right} \quad (2)$$

Now to put the model to use, GridSearch was put to use once more to find the following hyper-parameters: max depth and minimum samples per leaf. The ideal values, using a threefold cross-validation GridSearch, whereas follows max depth = 4 and minimum samples per leaf = 10. The mean cross-validation score of three folds is 0.8776. A confusion matrix in Fig.7 is computed to measure the prediction model performance.

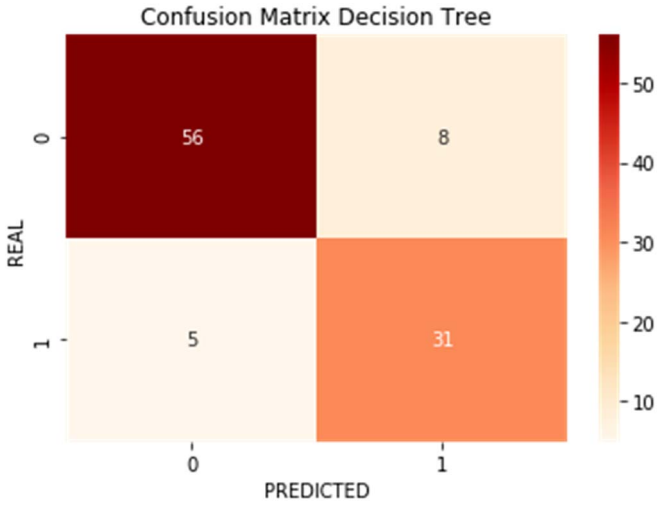

Fig. 7: Confusion matrix of the predicted values using Decision Trees

A significant performance measure is the ROC curve along with its AUC. The AUC ensures that the predictor focuses on the "FN"; this ensures that the predictor does not miss any eligible applicant. Fig 8 is a graph of the ROC of all three classifiers based on their decision functions; the trends of all the predictors look identical. A more accurate indicator of these curves is the AUC, which will be later calculated and tabulated.

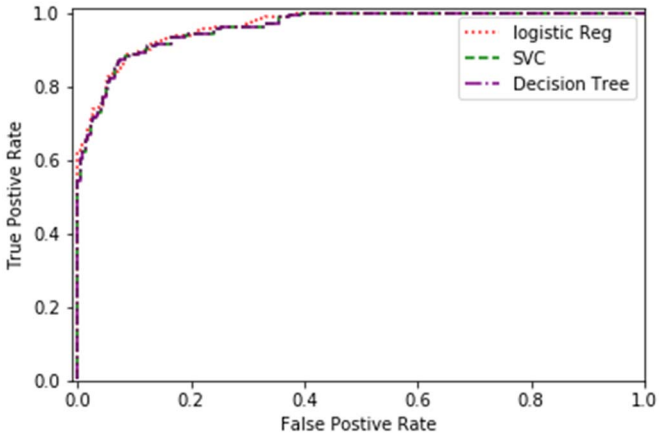

Fig.8: The ROC curves of all three classifiers

The performance measures accuracy, AUC, precision, and recall are all documented in Table 3. Where SVC and Logistic Regression gave identical and credible results; however, logistic regression outperformed SVC when it came to having a high recall rate without misidentifying the applicants that were accepted. The weakest classifier in terms of performance is the Decision Tree classifier; to improve the Decision Tree classifier performance a better variant of this classifier; Random Tree can be used instead.

TABLE 3: Performance measures of all classifiers

	Decision Tree	Logistic Regression	SVC
Accuracy	0.87	0.91	0.91
AUC	0.9251	0.9643	0.9613
Precision	0.7949	0.8461	0.8461
Recall	0.8611	0.9167	0.9167

C. *Ensemble Methods*

The initiative behind using ensembles is to improve the performance measure "accuracy" of weak classifiers: Ensemble methods work on employing several different or identical classifiers in parallel or sequentially. The principle behind using ensemble methods is that the accuracy is more likely enhanced. Four different Ensemble methods have been employed: Voting classifiers Bagging, AdaBoosting, and stacking.

The first ensemble method of use is the Voting Classifier, which was implemented on the dataset using previously implemented classifiers: Decision Tree, SVC, and Logistic Regression. The voting hyper-parameter chosen is Hard Voting: SVC does not provide any probability measure thus one is committed to using this hyper-parameter. The accuracy of the voting classifier is 0.91. This accuracy is intuitive; Logistic Regression and SVC are not only strong classifiers but also have a similar linear model classification technique in terms of a linear kernel; however, when comparing the accuracy of the voting classifier with that of the Decision Tree it does have improved accuracy.

Now for bagging, this Ensemble Method uses one classifier with randomly sampling repeated instances. Bagging

works by increasing the training data using bootstrap samples, this principle working technique is used for generalization. To ensure the bagging classifier has optimal performance a GridSearch was computed to obtain the following hyper-parameters max samples=0.6, max features =4, bootstrap =True and number of estimators =20. Bagging using Decision Trees improved the decision tree classifier's accuracy to a value of 0.9125, which is higher than the other three classifiers in terms of accuracy. Bagging can only effectively work on unstable classifiers that have a high tendency to over-fit.

Boosting, on the other hand, works differently; it places weights on each instance and each model, which in turn affects the performance of the next model by focusing on all the miss-classified instances. AdaBoost is the Boosting algorithm used in this research paper; AdaBoost helps with increasing variance in under fitted models. AdaBoost using decision Trees classifier is very effective in enhancing the estimator's accuracy; the downside to boosting is that the classifier can only work sequentially which is much more time-consuming.

Using GridSearch and importing AdaBoost from Scikit-learn the max number of estimators recommended is 10 at a learning rate of 0.85. Employing AdaBoost on Decision Trees, we get an accuracy of 0.9. Given that ten estimators are used each with an individual weight, a weight function is utilized to compute the weights of all ten classifiers and is in Fig.9. The initial estimators are more likely to have a higher weight given that Decision Trees split at estimators with better performance measures (Gini Index). Using AdaBoost on SVM classifiers is not effective due to them being originally very stable classifiers [7] the same could be said about Logistic Regression classifiers.

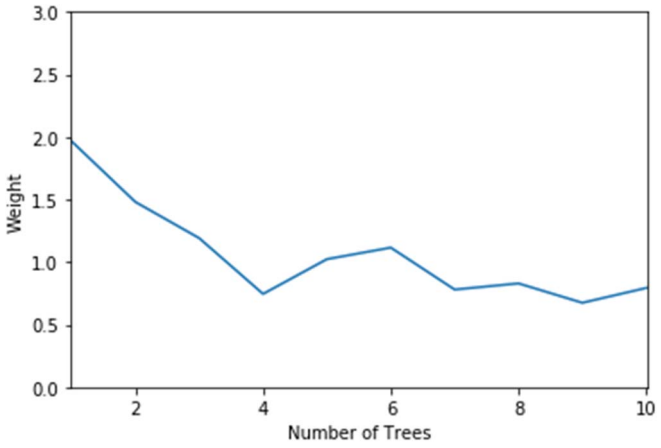

Fig.9: Weights of all ten trees in order

The final Ensemble method used is Stacking; Stacking is a hybrid between both Boosting and Bagging where the Stacking classifiers work with models in parallel as well as in sequence. Meta classifiers are employed to train on all the data predicted by the modeling layers ahead. Stacking is known for being the most effective Ensemble method in increasing the accuracy of several classifiers. The Stacking model is imported from MLXTEND library using four classifiers SVC, Random Forest, Gaussian Naive Bayes and as a meta-classifier Logistic Regression. After Training the Stacking classifier and then testing it on the test data, the accuracy of this classifier accuracy reached a value of 0.925. Which is considered superior amongst all classifiers along with all other ensemble methods in terms of accuracy.

V. Conclusion

Prediction of the graduate admission of applicants, where the dataset was made of eight important features related to educational history, using and comparing supervised machine learning models. This research documented several model performances whether it was the coefficient of determination for regression or accuracy and AUC for classification. Ensemble techniques were also implemented to improve accuracy especially in terms of weak or unstable classifiers. To implement this predictor using the given dataset any model considered in this research will provide a minimum accuracy of 0.87 and R^2 score of 0.724; however, using an Ensemble technique improved the accuracy score to an accuracy of 0.925. In general, all models can provide a good enough prediction given that the training data had double the number of accepted applicants to rejected applicants. This predictor proved to work competently; however, given the dataset, this predictor works only on the educational aspect of the applicant without taking into account personal data. This predictor has great potential in affecting the admission process worldwide.

The predictor presented can be implemented not only in university admission faculties [5] but also at recruiting agencies or human resources departments. Implementing this predicator would reduce the time needed to analyze the CVs of applicants. This would allow human resources to focus on applicants that are more legible.

References

[1] Chen, Y., Pan, C. C., Yang, G. K., & Bai, J. (2014, August). Intelligent decision system for accessing academic performance of candidates for early admission to university. In *2014 10th International Conference on Natural Computation (ICNC)* (pp. 687-692). IEEE.

[2] Hasan, M., Ahmed, S., Abdullah, D. M., & Rahman, M. S. (2016, May). Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)* (pp. 502-507). IEEE.

[3] Dietterich, T. G. (1997). Machine-learning research. *AI magazine*, *18*(4),

[4] Yazdipour, S., & Taherian, N. (2017, December). Data Driven Decision Support to Fund Graduate Studies in Abroad Universities. In *2017 International Conference on Machine Learning and Data Science (MLDS)* (pp. 44-50). IEEE.

[5] Waters, A., & Miikkulainen, R. (2014). Grade: Machine learning support for graduate admissions. *AI Magazine*, *35*(1), 64-64.

[6] Mohan S. Acharya, Asfia Armaan, Aneeta S Antony . A Comparison of Regression Models for Prediction of Graduate Admissions, 2019 IEEE International Conference on Computational Intelligence in Data Science

[7] Wickramaratna, J., Holden, S., & Buxton, B. (2001, July). Performancedegradation in boosting. In International Workshop on Multiple Classifier Systems (pp. 11-21). Springer, Berlin, Heidelberg.