Machine Learning and Deep Learning based Students' Grades Prediction

Korchi Adil (korchi.a@ucd.ac.ma)
Chouaib Doukkali University
Fayçal Messaoudi
Sidi Mohamed Ben Abdellah University
Abatal Ahmed
Hassan Premier University
Manzali Youness
Sidi Mohamed Ben Abdellah University

Research Article

Keywords: Machine learning, Predicting students' grades, Deep Neural Network, Regression, Supervised learning.

Posted Date: August 4th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3192793/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Operations Research Forum on October 31st, 2023. See the published version at https://doi.org/10.1007/s43069-023-00267-8.

Abstract

Predicting student performance in a curriculum or program offers the prospect of improving academic outcomes. By using effective performance prediction methods, instructional leaders can allocate adequate resources and instruction more accurately. This paper aims to identify the features of machine learning algorithms that can be used to make predictions about student grades. For this purpose, we use a data set that contains personal information about students and their grades. We have implemented different machine learning algorithms of regression namely: Decision Tree, Random Forest, Linear Regression, k-Nearest Neighbor, XGBoost, and Deep Neural Network. Then, we compared these models based on their determination coefficient, Mean Average Error, Mean Squared Error, and Root Mean Squared Error. The experimental results of this study showed that the deep neural network outperformed the other algorithms with a determination coefficient of 99.97% and low errors.

1. Introduction

Predicting students' marks is a common problem in educational data mining, with applications in areas such as student assessment, course design, and academic advising [1]. Machine learning techniques, such as linear regression and neural networks, can be used to build predictive models that can estimate students' marks based on various factors, such as past grades, attendance, and test scores. These models can help educators to identify at-risk students, design personalized learning interventions, and provide feedback to students on their progress [2].

There are various approaches that can be used to predict students' performance with machine learning algorithms, but they generally involve the following steps:

- Collect and prepare the data: This involves collecting the relevant data from students' records, such as their past grades, attendance, test scores, etc. To ensure that the data is in an acceptable format for the machine learning method, it should be cleaned and preprocessed.
- Choose a machine learning model: For this purpose, a variety of machine learning models, including neural networks, decision trees, and linear regression, can be utilized. The choice of model will depend on the characteristics of the data and the specific goals of the prediction.
- Train the model: Once the model has been chosen, it needs to be trained on the data. This involves feeding the model a large number of examples and changing the model's parameters to reduce the difference in scores between predictions and actual results.
- Evaluate the model: After the model has been trained, it is important to evaluate its performance to determine how well it is able to predict students' marks. This can be done using techniques such as cross-validation, where the model is tested on a portion of the data that was not used for training.
- Make predictions: The model can be used to make predictions on new data after it has been trained and assessed.

In this paper, we will discuss the various approaches that can be used to predict students' marks using machine learning, including data collection and preparation, model selection, model training, evaluation, and prediction.

The structure of this paper is as follows: Section 2 talks about some relevant research, and Section 3 lists the machine learning techniques that were employed, section 4 outlines the methodology adopted, section 5 shows the prediction results of the models used, and section 6 concludes the paper.

2. Related Work

Several studies have explored the use of machine learning for predicting student grades. One widely used approach is linear regression, which is a statistical method for finding the linear relationship between a dependent variable (in this case, student grades) and one or more independent variables (such as attendance, test scores and other factors). Linear regression has been shown to be effective in predicting student grades in a number of studies [3, 4, 5, 6].

Another popular approach for predicting student grades is the use of decision tree algorithms, which build a tree-like model of decisions based on the data. Decision trees have been used to predict student grades in a number of studies [7, 8, 9] and have been shown to be effective for this task.

In addition to linear regression and decision trees, other machine learning algorithms that have been used for predicting student grades include k-nearest neighbor (KNN) [10, 11], random forests [12]. These approaches have also been shown to be effective for this task, although they may have different strengths and limitations depending on the specific characteristics of the data and the goals of the prediction.

To predict students' performance based on the use of the internet as a learning resource and the impact of the time spent by students on social networks, the authors of a study [13] used a variety of machine learning algorithms, including decision trees, nave bayes, artificial neural networks (ANN), and logistic regression. They discovered that the ANN model, which had an accuracy of about 80%, performed the best.

The BiLSTM deep neural network model was employed by the authors in [14] coupled with an attention mechanism model, to predict students' grades from historical data. The results showed that the BiLSTM combined with the attention mechanism yielded a better accuracy of 90.16%.

In another study [15], the authors applied a deep learning model to predict students' academic performance. They employed a data set containing different variables such as demographic, social, educational, and student grades. They used the synthetic minority oversampling (SMOTE) technique to overcome the data imbalance problem. Their proposed solution resulted in approximately 96% accuracy for grade predictions across courses.

[16] looked into two data sets to predict and categorize student performance using several machine learning techniques, such as Backpropagation, Long-Short Term Memory, Support Vector Regression and for classification: Gradient Boosting Classifier. As a result, the Support Vector Regression model outperformed the other algorithms at the R-squared score of 83% in grade pediction, and for classification, the Backpropagation model performed the best with an accuracy equal to 87%.

3. Machine Learning Model used

- Decision Tree Regressor

A decision tree is a machine learning method used to categorize data or make predictions based on the answers provided to a series of previous questions. This model is a type of supervised learning, which means that it is trained and tested on a data set with the required categorization. It is a graphical representation that provides all possible solutions to a problem from given conditions.

- Random Forest Regressor

A supervised learning technique [17] [18] called a random forest employs an ensemble learning approach for regression. It is a meta-estimator that employs the mean to increase prediction accuracy and reduce overfitting. It does this by fitting a number of classification decision trees to different subsamples of the data set [19].

- Linear Regression

Linear regression is a popular statistical method used for modeling the relationship between a dependent variable and one or more independent variables. It is a simple but powerful technique that assumes a linear relationship between the variables; its purpose is to solve regression problems. Regression builds a target prediction value on a set of independent variables. Linear regression is principally employed to find the relationship between variables and predictions.

- k-Nearest Neighbors Regressor

k-Nearest Neighbors (k-NN) Regressor is a type of supervised learning algorithm used for regression tasks. It works on the principle of finding the k-nearest data points to a new, unseen data point and using their target values (dependent variable) to predict the value for the new data point. Here's how the k-NN Regressor algorithm works :

- Training: During the training phase, the algorithm stores the feature vectors and their corresponding target values (dependent variable) from the training dataset.
- Prediction: When given a new, unseen data point for which we want to predict the target value

- XGBoost Regressor

XGBoost is a supervised machine learning algorithm used on large data sets. It is an accurate implementation of gradient boosting which can be applied to predictive modeling by regression.

- Deep Neural Network

A Deep Neural Network is characterized by a particularity that it is composed of an input layer, an output layer and at least 3 layers in between of interconnected nodes, or "neurons.". This allows it to process data in a complex way, using advanced mathematical models. Each of these layers performs different types of sorting and specific categorization in a process called feature hierarchy.

4. Methodology

The methodology of this study includes the following steps, which are summarized in the figure below (Fig. 1).

4.1. Data set description

The goal of this study is to predict students' total scores using techniques of machine learning and deep learning. To achieve this, we used a data set containing information on various 1000 student characteristics, including gender, of education level of parents, lunch, and exam preparation courses. In addition, the data set included scores on math, reading, and writing exams as shown in Fig. 2.

4.2. Data cleaning and preprocessing:

The first step in our analysis was to clean and preprocess the data. This included handling missing values, converting categorical variables to numeric form (Fig. 4), and scaling the data to ensure that all variables were on the same scale.

4.3. Feature engineering:

To improve the performance of the machine learning algorithms, we performed feature engineering on the data set. This involved selecting the most relevant features and creating new feature: Total score, the target variable, by combining the existing ones (math, reading, and writing scores) as shown below (Fig. 5). Feature engineering seeks to produce more robust and predictive data set for the machine learning algorithms (Fig. 6).

4.4. Data Visualization

We then visualized the data to understand the data set and the correlations between different variables and to identify any patterns or trends. This helped us to identify the most important features for predicting students' total scores. Figures 7, 8, 9, and 10 represent graphically the different variables.

A correlation study is a statistical analysis that measures the relationship between two or more variables. It is used to understand how the values of one variable are affected by changes in the values of another variable. The degree and direction of the linear link between variables is measured by the Pearson correlation coefficient. Its value falls between − 1 and 1, with − 1 denoting a high negative correlation, 0 denoting no correlation, and 1 denoting a significant positive correlation. To perform a correlation study in Python, the corr() method of a Pandas DataFrame or the pearsonr() function from the scipy.stats module can be used.

Figure 11 shows that there is a strong correspondence between the variables: math score, reading score, writing score, and the target variable Total score.

4.5. Data splitting:

To ensure the validity of our results, we used a 30/70 ratio to divide the data set into training and testing sets. The testing set was used to gauge how well the models performed, while the training set was used to train the machine learning algorithms.

4.6. Machine learning algorithms implementation:

We then implemented a range of machine learning algorithms for predicting student grades, including decision trees, random forests, linear regression, k-nearest neighbor, XGBoost, and deep neural networks [20].

The procedures listed below were used to implement these algorithms using the scikit-learn library and the Python programming language.

- Create an instance of the algorithm class: Each algorithm is implemented using a corresponding class from a library such as scikit-learn. To create an instance of the class, we need to call the class with any desired hyperparameters (the random_state parameter is set to ensure that the results are reproducible).
- Fit the model to the training data: The fit() method is used to fit the model after it has been generated to the training set of data. This method takes the training data and target variables as inputs and adjusts the model's internal parameters to fit the data.
- Repeat the process for each algorithm: The process of creating and fitting the model is repeated for each algorithm. After all the algorithms are trained, they can be used to make predictions on the testing data.

4.7. Deep Neural Network implementation:

The DNN used was built using the Keras library in Python using the following steps:

The first step is to create a model object using the Sequential class. This creates a model that is a linear stack of layers, where the input goes through each layer sequentially and the output of one layer is the

input of the next layer.

Next, the layers are added to the model using the model.add() method. The model has 15 layers, each with a specified number of neurons and an activation function. The activation function determines the output of a neuron given an input or set of inputs. In this case, the relu activation function is used for all but the output layer, which uses a linear activation function.

After the layers are added, the model is compiled using the model.compile() method. This step specifies the optimizer, loss function, and metrics that will be used to train the model. The Adam optimizer is used, and the loss function is the mean squared error (MSE). The MAE, MSE, and RMSE metrics are also used to evaluate the model's performance.

Finally, the model is trained using the model.fit() method, which takes the training data and target variables as inputs and trains the model for the given number of epochs. The model is trained for 100 epochs. The model is also evaluated on the testing data using the validation data parameter.

5. Model Evaluation and Results

To assess and contrast how well the machine learning algorithms work, we used regression plots of the models as well as several metrics, including the R-squared score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

5.1 Regression plots:

A regression plot is a scatter plot that illustrates the connection between two variables and the fitted line or curve that represents the model's predictions. It is a useful tool for visualizing the performance of a machine learning model and identifying trends and patterns in the data.

We used regression plots to visualize the predictions made by each model based on the test set (y_test). These plots showed the relationship between between expected and actual values, allowing us to see how well each model was able to accurately predict the total scores of students as shown in the following figures (Fig.12, 13, 14, 15, 16, 17). The regression graphs can be created in Python by the regplot() function of the Seaborn library.

Based on the regression plots of the models used. We can see that the DNN model fits the data better.

5.2 Performance metrics:

- **R-squared score:** This metric gauges how much of the target variable's variance the model is able to account for. An improved fit is indicated by a higher R-squared value.
- **Mean absolute error (MAE):** It is the average absolute difference between the values that were predicted and the actual values is measured by the mean absolute error (MAE). Better fit is indicated by a lower MAE.

- **Mean squared error (MSE):** The average squared difference between the anticipated values and the actual values is measured by the mean squared error (MSE). Better fit is indicated by a lower MSE.
- Root mean squared error (RMSE): This measure is used to quantify the error in the same units as the target variable and is the square root of the MSE. Better fit is indicated by a lower RMSE.

The table below shows the R-squared, MAE, MSE, and RMSE values of the different models used (Tab.1).

Table 1. Model evaluation based on R-squared, MAE, MSE, and RMSE

Model	R-squared (%)	MAE	MSE	RMSE
DT	97.32	3.17	17.24	4.15
LR	99.10	3.42	1.81	4.26
RF	98.59	1.44	4.00	2.00
k-NN	98.72	1.39	3.44	1.85
XGB	96.04	6.02	49.84	7.06
DNN	99.97	0.45	0.05	1.13

Based on the results of the regression plots and the different evaluation metrics shown in table.1, the DNN model was found to be the best performing model with a determination coefficient equal to 99.97% and MAE= 0.45, MSE= 0.05, RMSE=1.13. Followed by the LR model with an R-squared equal to 99.10% and relatively high errors. In third position there is the k-NN, followed by the RF model, then the DT, and the XGB in last position.

6. Conclusion

The objective of this paper is to apply machine learning algorithms for the prediction of student scores. After implementing and evaluating a range of machine learning algorithms, our findings demonstrated that the deep neural network model performed better than the competing algorithms in terms of determination coefficient and error metrics. With a determination coefficient of 99.97% and negligible errors, the deep neural network demonstrated the highest level of accuracy in predicting students' grades.

These results have important implications for educators and administrators looking to use machine learning to improve student outcomes and support student success. By identifying the most effective algorithms for predicting student grades, we can better understand the factors that contribute to student performance and tailor teaching approaches and support to the specific needs of individual students.

Overall, this study highlights the potential of machine learning to revolutionize the way we approach education by providing personalized and targeted support to students. By continuing to explore and refine these techniques, we can continue to make progress in helping students achieve their full potential.

Declarations

Funding : The authors did not receive support from any organization for the submitted work.

Conflict of Interest : The author states that there is no conflict of interest.

Ethical approval : Not Applicable.

Consent to participate : Not Applicable.

Consent for publication : All authors of the manuscript have agreed for authorship, read and approved the manuscript, and given consent for the submission of the manuscript.

Data availability : The datasets used during the current study are freely available in the UCI repository.

Code availability : The code will be available upon request to reviewers.

Authors Contribution

The authors confirm their contribution to the paper as follows:

- Study conception and design: Korchi Adil, Messaoudi Fayçal, Manzali Youness, Abatal Ahmed.
- Data collection: Korchi Adil, Messaoudi Fayçal, Manzali Youness, Abatal Ahmed.
- Analysis and interpretation of results: Korchi Adil, Messaoudi Fayçal, Manzali Youness, Abatal Ahmed.

Draft manuscript preparation: Korchi Adil, Messaoudi Fayçal, Manzali Youness, Abatal Ahmed.

All authors reviewed the results and approved the final version of the manuscript.

References

1. Okewu, E., Adewole, P., Misra, S., Maskeliunas, R., & Damasevicius, R. (2021). Artificial neural networks for educational data mining in higher education: A systematic literature review. Applied Artificial Intelligence, 35(13), 983-1021.
2. Bañeres, D., Rodríguez, M. E., Guerrero-Roldán, A. E., & Karadeniz, A. (2020). An early warning system to detect at-risk students in online higher education. Applied Sciences, 10(13), 4427.
3. Yang, S. J., Lu, O. H., Huang, A. Y., Huang, J. C., Ogata, H., & Lin, A. J. (2018). Predicting students' academic performance using multiple linear regression and principal component analysis. *Journal of Information Processing*, *26*, 170-176.
4. Huang, S., & Fang, N. (2013). Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. *Computers & Education*, *61*, 133-145.

5. Gorr, W. L., Nagin, D., & Szczypula, J. (1994). Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, *10*(1), 17-34.
6. Gadhavi, M., & Patel, C. (2017). Student final grade prediction based on linear regression. *Indian J. Comput. Sci. Eng*, *8*(3), 274-279.
7. Al-Barrak, M. A., & Al-Razgan, M. (2016). Predicting students final GPA using decision trees: a case study. *International journal of information and education technology*, *6*(7), 528.
8. Kolo, D. K., & Adepoju, S. A. (2015). A decision tree approach for predicting students academic performance.
9. Hamoud, A., Hashim, A. S., & Awadh, W. A. (2018). Predicting student performance in higher education institutions using decision tree analysis. *International Journal of Interactive Multimedia and Artificial Intelligence*, *5*, 26-31.
10. *Amra, I. A. A., & Maghari, A. Y. (2017, May). Students performance prediction using KNN and Naïve Bayesian. In 2017 8th International Conference on Information Technology (ICIT) (pp. 909-913). IEEE.*
11. Maghari, A. (2018). Prediction of student's performance using modified KNN classifiers. In *Alfere, SS, & Maghari, AY (2018). Prediction of Student's Performance Using Modified KNN Classifiers. In The First International Conference on Engineering and Future Technology (ICEFT 2018)* (pp. 143-150).
12. Batool, S., Rashid, J., Nisar, M. W., Kim, J., Mahmood, T., & Hussain, A. (2021, July). A random forest students' performance prediction (rfspp) model based on students' demographic features. In *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)* (pp. 1-4). IEEE.
13. Altabrawee, H., Ali, O. A. J., & Ajmi, S. Q. (2019). Predicting students' performance using machine learning techniques. *JOURNAL OF UNIVERSITY OF BABYLON for pure and applied sciences*, *27*(1), 194-205.
14. Yousafzai, B. K., Khan, S. A., Rahman, T., Khan, I., Ullah, I., Ur Rehman, A., ... & Cheikhrouhou, O. (2021). Student-performulator: student academic performance using hybrid deep neural network. *Sustainability*, *13*(17), 9775.
15. Aslam, N., Khan, I., Alamri, L., & Almuslim, R. (2021). An Improved Early Student's Academic Performance Prediction Using Deep Learning. *International Journal of Emerging Technologies in Learning (iJET)*, *16*(12), 108-122.
16. Sekeroglu, B., Dimililer, K., & Tuncal, K. (2019, March). Student performance prediction and classification using machine learning algorithms. In *Proceedings of the 2019 8th International Conference on Educational and Information Technology* (pp. 7-11).
17. Burdakov, O. (2020, March). Ioannis C. Demetriou and Panos M. Pardalos (eds): Approximation and Optimization: Algorithms, Complexity and Applications. In SN Operations Research Forum (Vol. 1, pp. 1-5). Springer International Publishing.
18. Korani, W., & Mouhoub, M. (2021, September). Review on nature-inspired algorithms. In Operations research forum (Vol. 2, pp. 1-26). Springer International Publishing.

19. Manzali, Y., & Elfar, M. (2023, June). Random Forest Pruning Techniques: A Recent Review. In Operations Research Forum (Vol. 4, No. 2, pp. 1-14). Springer International Publishing.
20. Mitra, A., Jain, A., Kishore, A., & Kumar, P. (2022, September). A comparative study of demand forecasting models for a multi-channel retail company: a novel hybrid machine learning approach. In Operations Research Forum (Vol. 3, No. 4, p. 58). Cham: Springer International Publishing.

Figures

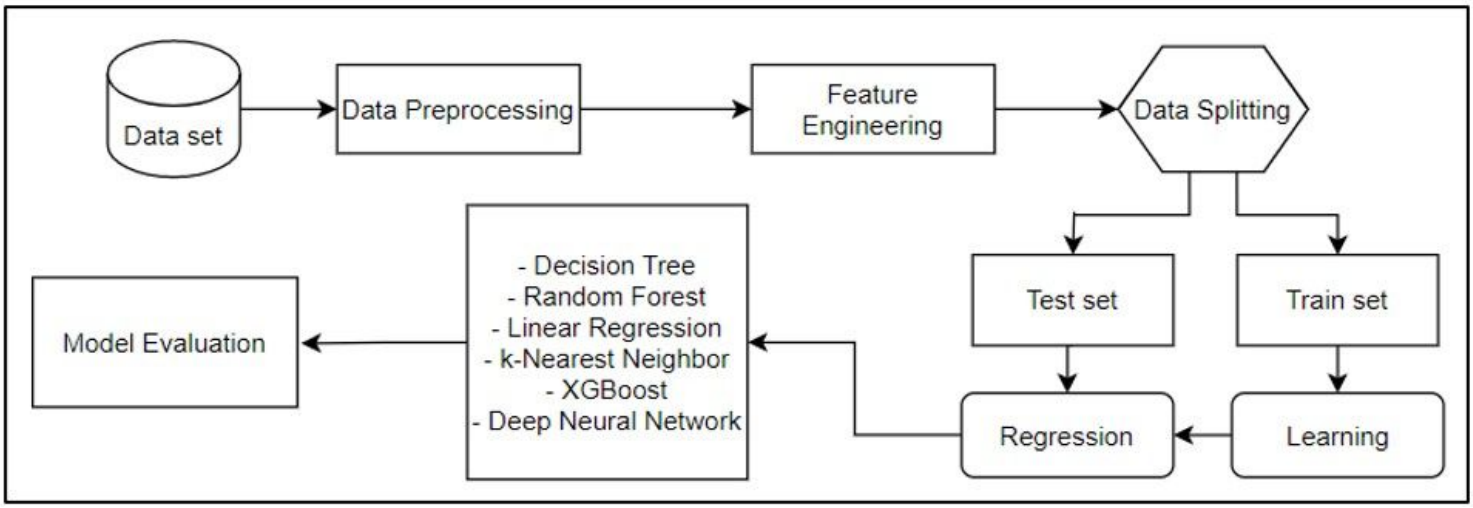

Figure 1

The study's pipeline

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
5	female	group B	associate's degree	standard	none	71	83	78
6	female	group B	some college	standard	completed	88	95	92
7	male	group B	some college	free/reduced	none	40	43	39
8	male	group D	high school	free/reduced	completed	64	64	67
9	female	group B	high school	free/reduced	none	38	60	50
10	male	group C	associate's degree	standard	none	58	54	52

Figure 2

Data set used

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

Figure 3

Data set information

	gender_male	race/ethnicity_group B	race/ethnicity_group C	race/ethnicity_group D	race/ethnicity_group E	parental level of education_bachelor's degree	parental level of education_high school	parental level of education_master's degree	parental level of education_some college	parental level of education_some high school	lunch_standard	test preparation course_none
0	0	1	0	0	0	1	0	0	0	0	1	1
1	0	0	1	0	0	0	0	0	1	0	1	0
2	0	1	0	0	0	0	0	1	0	0	1	1

Figure 4

Categorical variables conversion

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total score
0	female	group B	bachelor's degree	standard	none	72	72	74	218
1	female	group C	some college	standard	completed	69	90	88	247
2	female	group B	master's degree	standard	none	90	95	93	278
3	male	group A	associate's degree	free/reduced	none	47	57	44	148
4	male	group C	some college	standard	none	76	78	75	229
5	female	group B	associate's degree	standard	none	71	83	78	232
6	female	group B	some college	standard	completed	88	95	92	275
7	male	group B	some college	free/reduced	none	40	43	39	122
8	male	group D	high school	free/reduced	completed	64	64	67	195
9	female	group B	high school	free/reduced	none	38	60	50	148
10	male	group C	associate's degree	standard	none	58	54	52	164

Figure 5

Creation of the target variable “Total score”

	gender_male	race/ethnicity_group B	race/ethnicity_group C	race/ethnicity_group D	race/ethnicity_group E	parental level of education_bachelor's degree	parental level of education_high school	parental level of education_master's degree	parental level of education_some college	parental level of education_some high school	lunch_standard	test preparation course_none	math score	reading score	writing score	Total score
0	0	1	0	0	0	1	0	0	0	0	1	1	72	72	74	218
1	0	0	1	0	0	0	0	0	1	0	1	0	69	90	88	247
2	0	1	0	0	0	0	0	1	0	0	1	1	90	95	93	278

Figure 6

Final data set

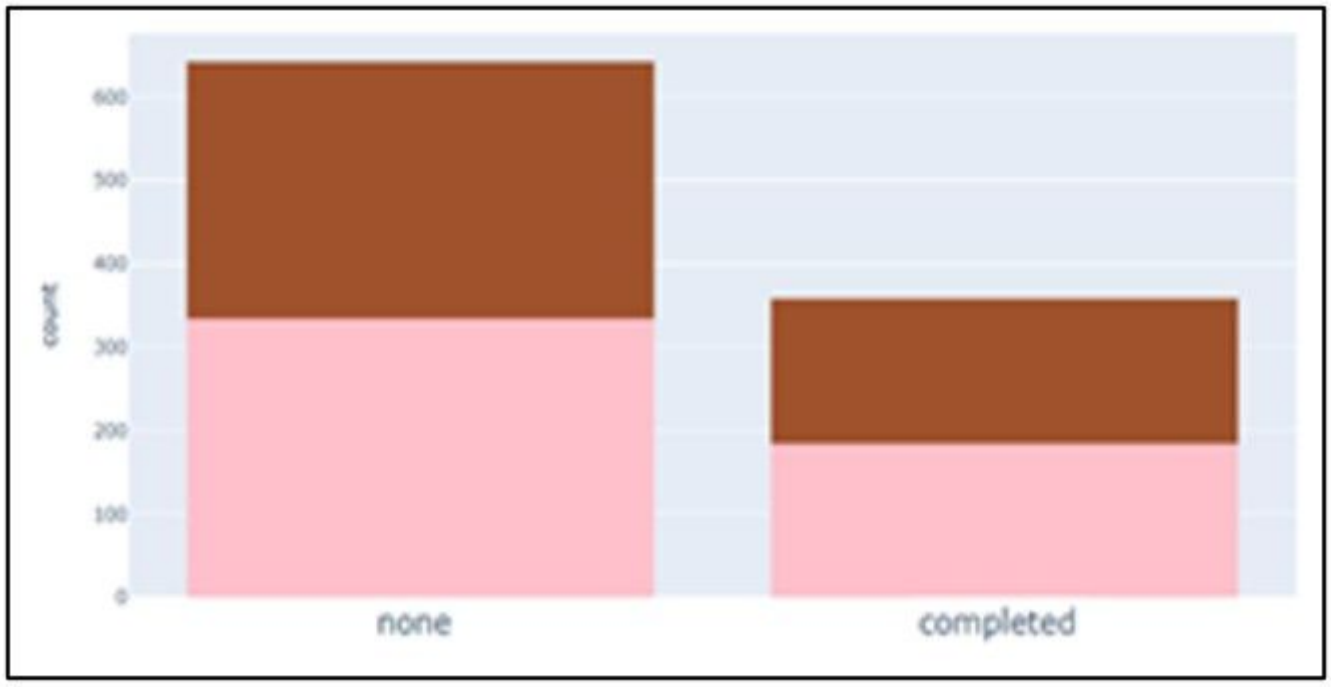

Figure 7

Distribution of the variable "gender" according to "test preparation course" (completed or not)

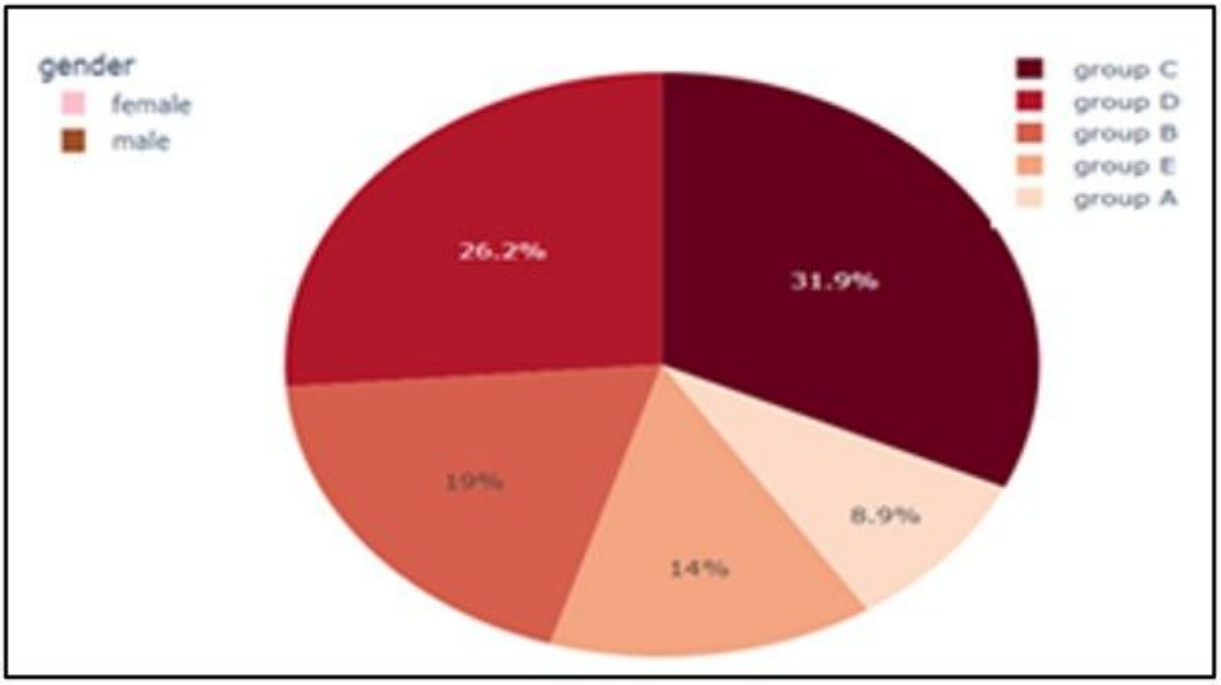

Figure 8

Distribution of the variable "race/ethnicity

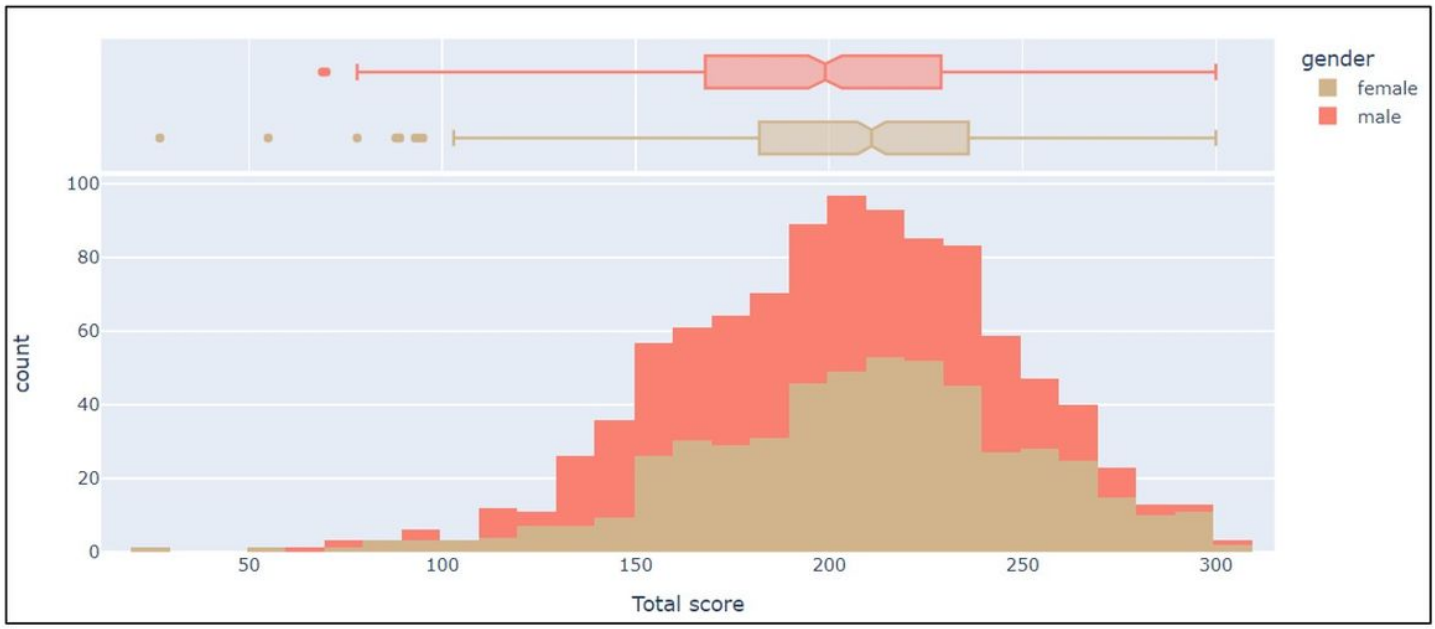

Figure 9

Distribution by gender of the total score

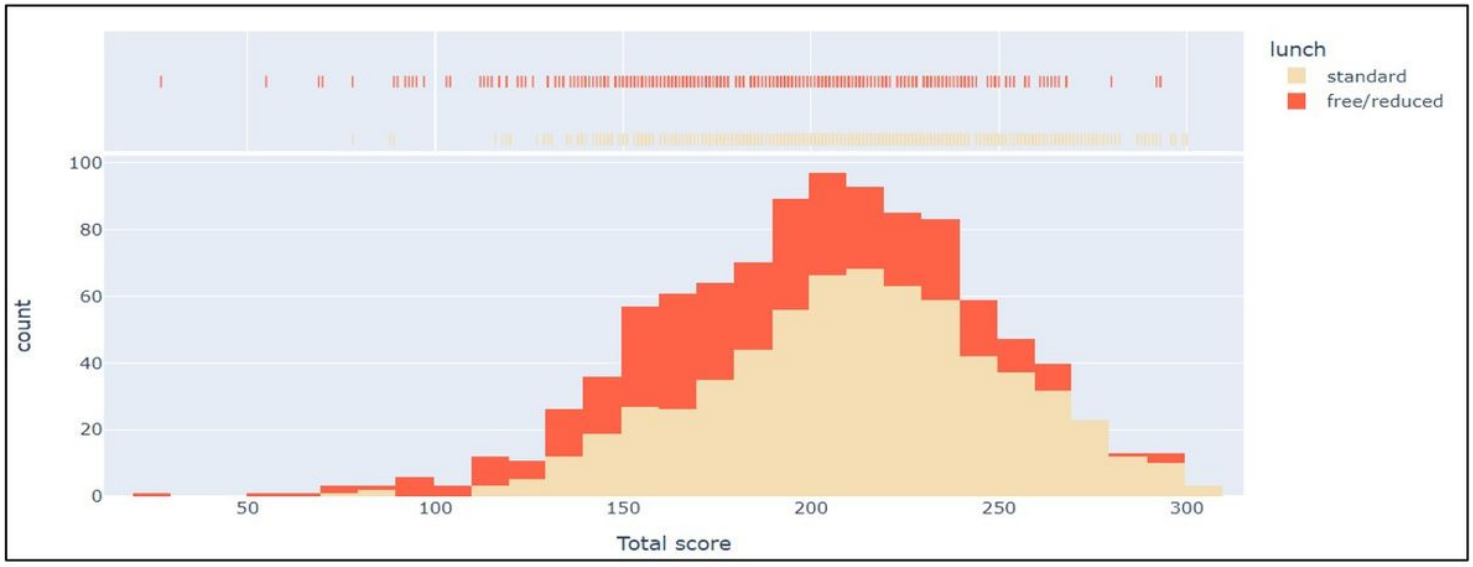

Figure 10

Distribution of total scores by lunch type

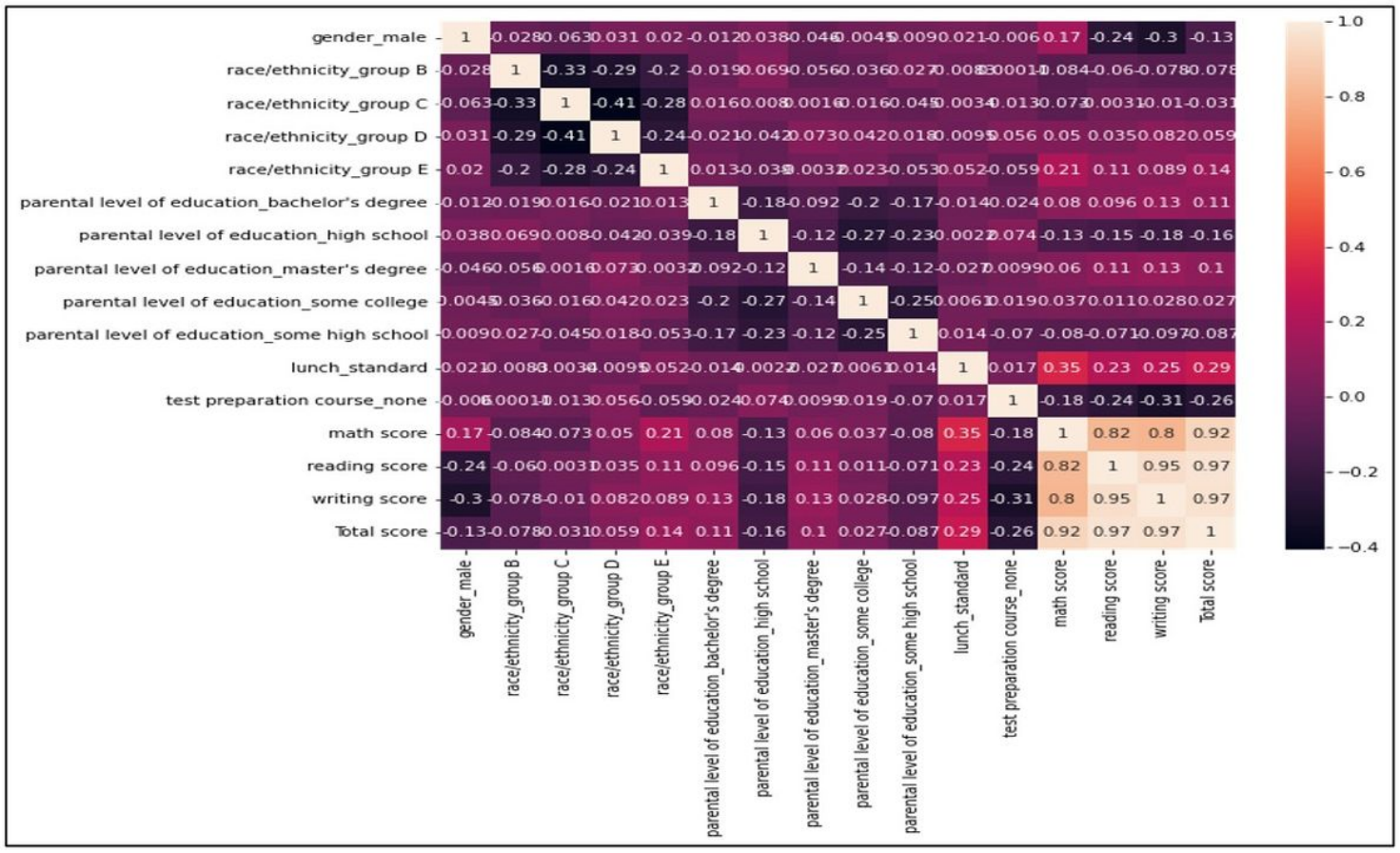

Figure 11

Correlation study

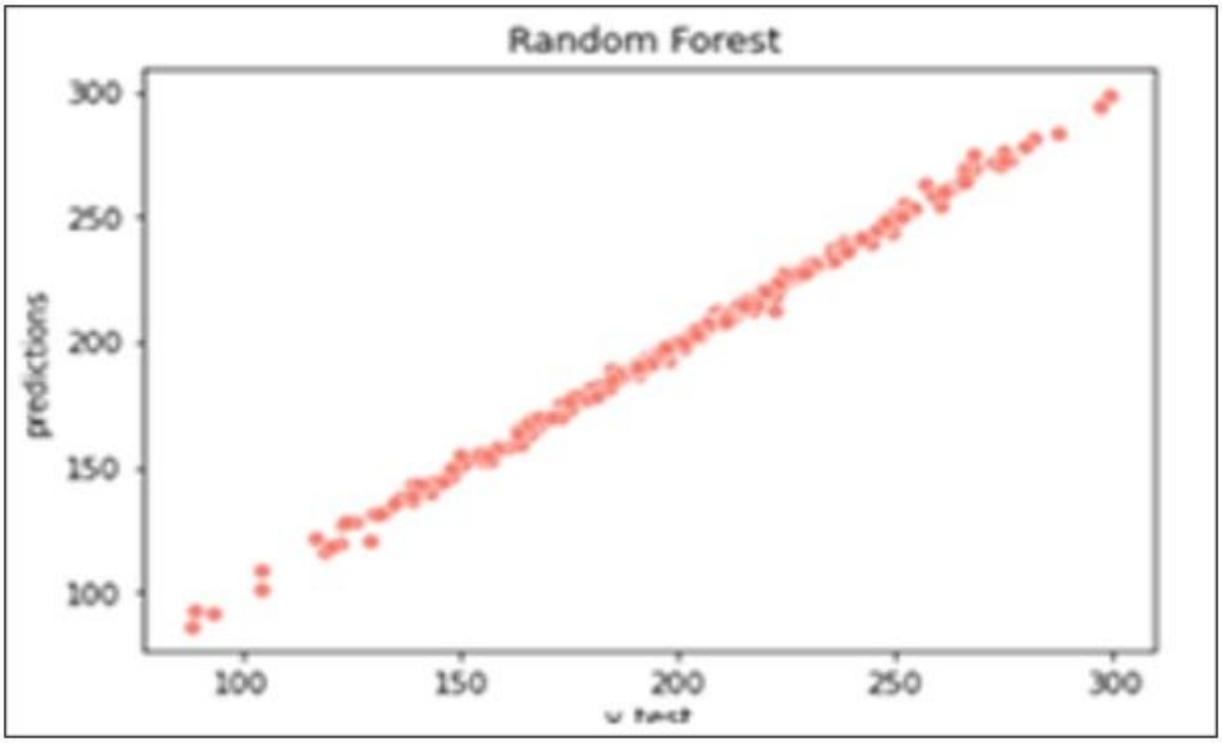

Figure 12

Random Forest regression plot

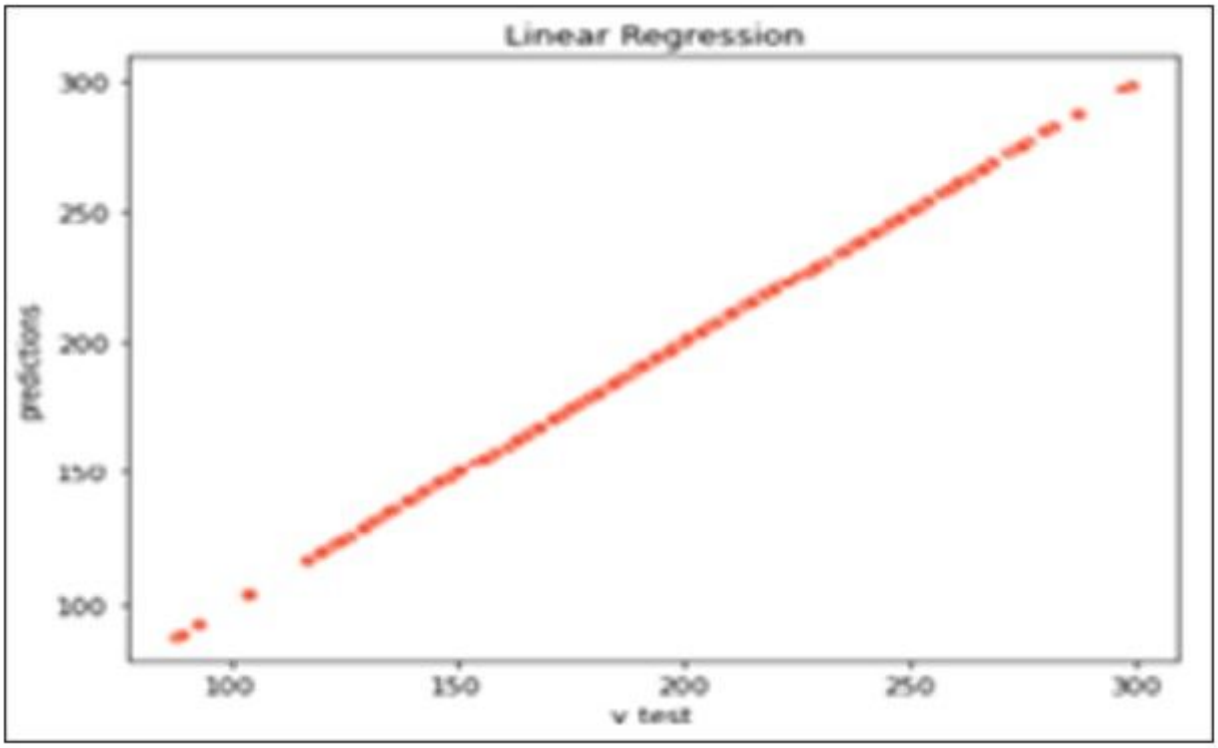

Figure 13

Linear Regression regression plot

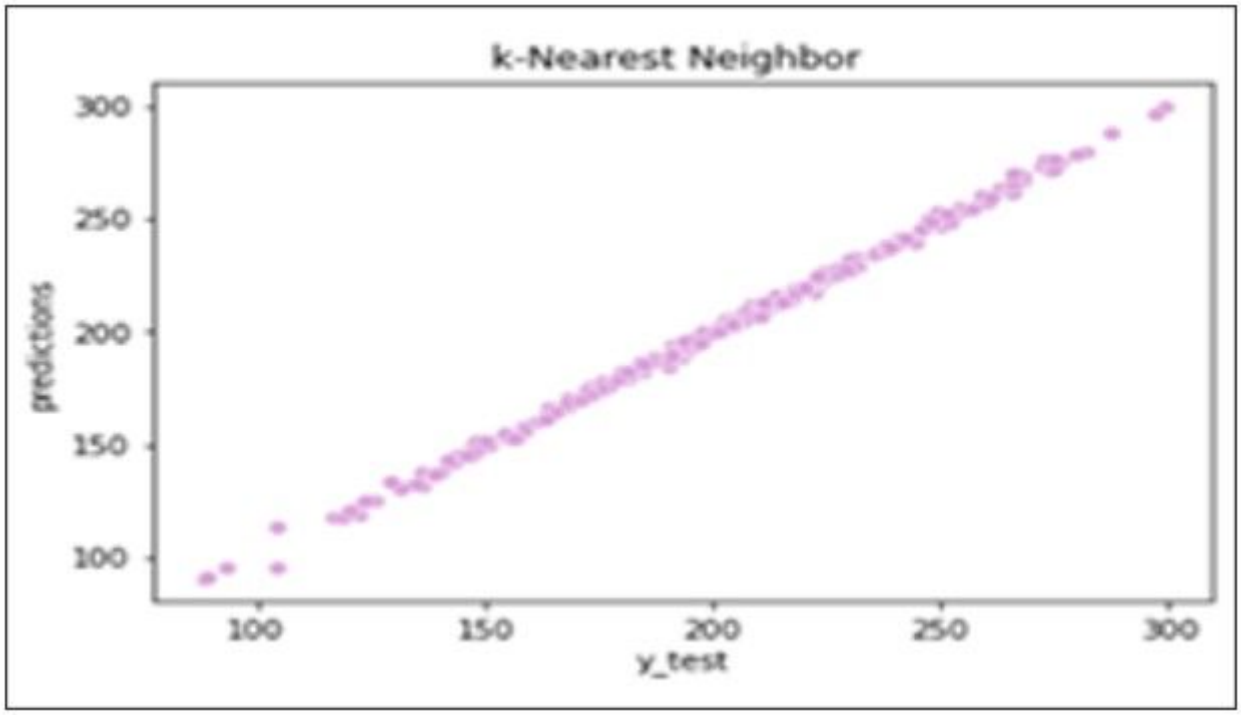

Figure 14

k-Nearest Neighbors regression plot

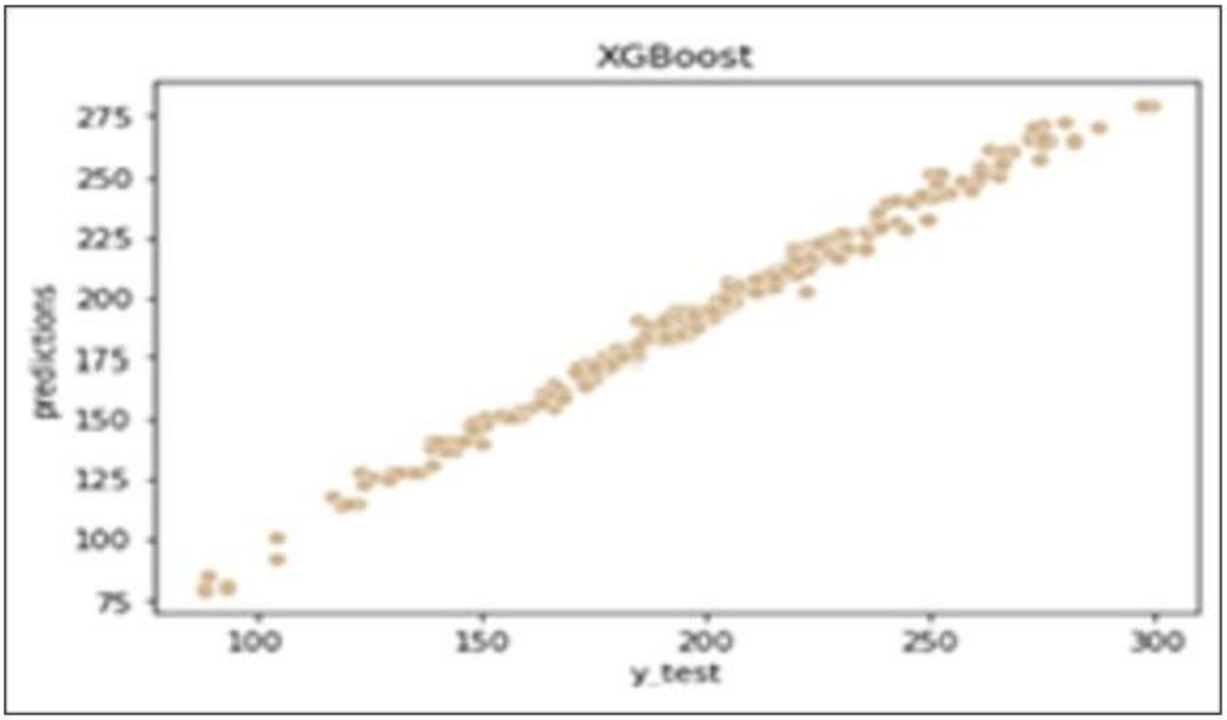

Figure 15

XGBoost regression plot

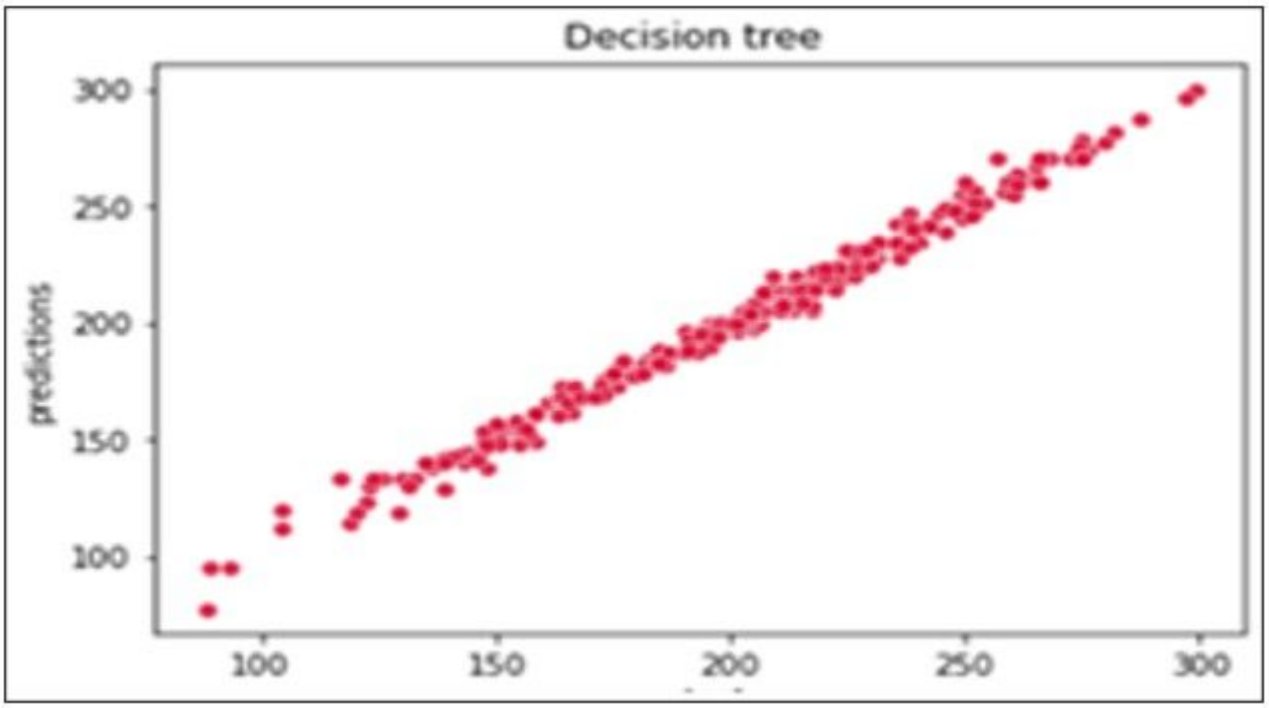

Figure 16

Decision Tree regression plot

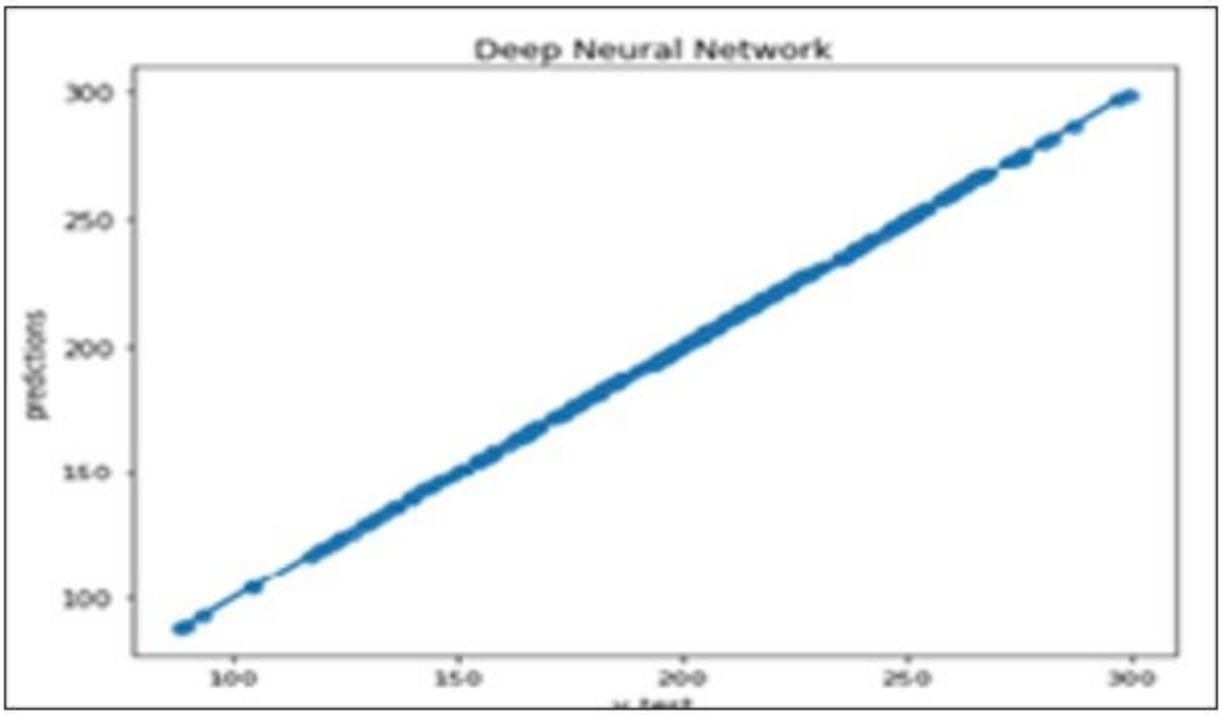

Figure 17

Deep Neural Network regression plot