* Title
* (A concise and descriptive title that reflects the main focus of the research.)
* FirstName LastName1,, Dr. A.D. Sawarkar1\*, Third2
* *1*Department of Information Technology, Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIET)*, Nanded*
* *2*Department of, College*, City*
* *adsawarkar@sggs.ac.in1\**
* **Abstract**:It is very important to use the right prediction tool to predict the **percentile** of students entering engineering colleges. This research addresses this challenge by providing powerful predictive  models using machine learning, specifically decision tree regression models. Thanks to the  intuitive interface created using Python widgets, users can reach accurate prediction results by  entering important information such as gender, category, seat type, school name and branch. The model has been made reliable and accurate by working on widely accepted historical data.

Compared to other models, it demonstrates the superiority of the decision tree regressor with performance metrics such as squared error and Rsquared score. The interface serves as a transparent and userfriendly interface to help potential students and admissions teams make informed decisions.

Study participants provided insight. By providing an accurate percentage of the prediction, this research helps improve the fairness and efficiency of the admissions process.

* Keywords: Regression Model, decision tree regressor, Ridge regressor, Random forest and Knearest regressor, admission to engineering colleges, user interface.
* 1. Introduction:
* In today's competitive business environment, it is important for students who want to be successful in their careers to attend prestigious schools. This is especially important for engineering jobs where the demand for technical skills is high. However, the complexity of the admission process creates a huge challenge for these students, especially in a state like Maharashtra where there are many engineering colleges and branches. is the gateway to engineering colleges, but with over 1.5 million seats spread across 200+ colleges and 35+ branches, sorting through the options is daunting. Students should carefully consider the list of possible universities based on their studies and interests, but relationships and competition often lead to a poor choice. A computer-based approach to improving college choice. Using academic data and user preferences, our tool can predict the best university for each student, including factors such as course selection, location and exclusion criteria. List of institutions that will support students with their needs, enable them to make informed decisions and, above all, increase their chances of admission to universities. By studying and simplifying the admissions process, we aim to reduce the risks associated with university choice and make the transition to higher education easier.

2. Literature Review:

Machine learning regression has become a powerful tool in predictive modeling that provides insight into complex data and facilitates decision-making in many areas. Many studies have investigated different algorithms, methods, and machine learning regression techniques. Horizontal regression, ridge regression, decision tree regression, KNN regression and random forest regression are the main techniques used in predictive modeling. Modeling relationships between linear equations. Ridge regression is a type of linear regression that alleviates multicollinearity problems by introducing a time penalty. Decision tree regression divides the feature space into regions and predicts the response based on the average of training events in each region. KNN regression estimates the response deviation from the mean of the values ​​of its nearest neighbors. Random forest regression is an integrated method that combines multiple decision trees to improve prediction accuracy and robustness. , but there are still some gaps and flaws in the data. One important difference is that there is no good model that includes the many factors that influence college admissions, such as academics, course preferences, and geographic preferences. Existing models often focus on the characteristics of forecasting product competition and do not capture all the challenges of entry. Preferences and limitations. Existing systems may be opaque or require expertise, limiting student access. It's a process to address these gaps. By combining advanced machine learning techniques and using multiple predictive models, the model aims to provide personalized and personalized recommendations, thus enabling students to make informed decisions about their education.

* 3. Methodology:
* 3.1Data Cleaning:  
  Data was adjusted from “mht\_cet2.csv” and cleaned to resolve missing values, discrepancies, and inconsistencies. Missing data are imputed using appropriate techniques (such as mean, median, or mode imputation), or rows with missing values ​​are removed entirely to maintain data integrity. The results are processed using methods such as truncation, separation, or statistical measures to reduce their impact on the model. Data sets are compatible. average 1. This step ensures that all features contribute equally to the training sample and avoids bias due to small differences. The model references data to predict additional functions. This may include creating correlation coefficients, multivariate specifications, or aggregating data from different sources to provide better input for regression models. ", 'Secondary seat type', 'score\_type', 'college\_name' and 'branch' are encoded using techniques such as single cube coding or label coding to convert them into a numerical format package for model training. transformation:  
  Data transformation technology, regression It uses logarithmic or power transformation and other methods to resolve the skewed distribution and nonlinear relationship between features and target variable data to meet the model's assumptions and improve its performance.
* 3.2 Algorithm Selection and Model Training: Here we split the data into training and testing sets and train the model on chosen algorithms on the training data. We used four test sizes: 0.2 The selection of a regression algorithm depends on the nature of the data and the problem statement. In this project, we have multiple independent and a single dependent variable (Percentile ). To choose the best regression algorithm, we can follow these steps:
* 1.Linear Regression: We can start by using a simple linear regression model to see how well the data fits the model. Linear regression assumes a linear relationship between the independent and dependent variables. If the data has a linear relationship, linear regression can be a good choice.
* 2. Ridge Regression:
* Ridge regression is a regularization technique used to mitigate overfitting in linear regression models by adding a penalty term to the loss function. It is particularly useful when the dataset has multicollinearity, where independent variables are highly correlated with each other.
* 3.KNN Regressor: KNN Regressor is a machine learning algorithm used for regression tasks. It predicts the target value of a new data point based on the average of the knearest neighbors target values in the training set, where k is a user-defined hyperparameter.
* 4.Decision Tree Regressor: If the data has complex and non-linear relationships, decision tree regression can be a good choice. Decision tree regression can capture complex relationships between the independent and dependent variables. 4.Random Forest Regressor: Random forest regression is a popular regression algorithm that uses multiple decision trees to make predictions. It can handle non-linear relationships between the independent and dependent variables and can also prevent overfitting.
* 3.3.OVERVIEW OF ALGORITHMS USED
* 3.3.1 LINEAR REGRESSION
* A form of regression technique called linear regression uses one or more input variables, usually referred to as independent variables or features, to predict a continuous target variable. A straight line can be used to show the connection between the input variables and the target variable in linear regression models. The target variable's predicted and actual values are compared using the linear regression procedure to identify the line of best fit that minimizes the sum of the squared residuals.
* An equation in the form of: represents the line of best fit.
* y = β0 + β1x1 + β2x2 + ... + βnxn
* where y is the predicted value of the target variable, β0 is the intercept or bias term, β1, β2, ..., βn are the coefficients of the input variables x1, x2, ..., xn, respectively. The linear regression technique employs the Ordinary Least Squares (OLS) method, which minimizes the sum of the squared residuals, to get the ideal coefficient values. The disparities between the target variable's expected and actual values are known as the residuals.By entering the values of the input variables into the equation after the coefficients have been computed, we may utilize the linear regression model to predict new data. A popular and straightforward regression approach that is simple to understand and use is linear regression. The assumption of linearity, vulnerability to outliers, and inability to manage non-linear connections between the input and target variables are a few of its drawbacks.
* 3.3.2 Ridge Regression:
* Ridge regression is a regularization technique used to mitigate overfitting in linear regression models by adding a penalty term to the loss function. It is particularly useful when the dataset has multicollinearity, where independent variables are highly correlated with each other.
* Overview:
* Ridge regression extends linear regression by adding a regularization term to the ordinary least squares (OLS) method. This regularization term, also known as the L2 penalty, penalizes large coefficients by adding their squares to the loss function. As a result, ridge regression shrinks the coefficients toward zero, reducing their variance and mitigating overfitting.
* Methodology:
* Model Training: Similar to linear regression, ridge regression is trained on the training data to learn the coefficients that minimize the combined error of the loss function and the regularization term.
* Regularization Parameter lambda: The strength of regularization is controlled by a hyperparameter called- lambda. A higher value of lambda increases the regularization strength, leading to more shrinkage of coefficients.
* Model Evaluation: After training, the ridge regression model is evaluated using the same metrics as linear regression, such as Mean Squared Error (MSE) and R-squared score.
* Data Preprocessing:
* Ridge regression follows similar preprocessing steps as linear regression, including data cleaning, normalization, and feature engineering. However, it is particularly effective when dealing with multicollinearity, as it helps stabilize the coefficient estimates.
* By incorporating ridge regression into the modeling process, we can effectively handle multicollinearity and improve the generalization performance of the model.
* 3.3.3 KNN REGRESSOR:
* The K-Nearest Neighbors (KNN) Regressor is a kind of regression method that makes predictions based on the separations between the input data points. As the KNN regressor is a non-parametric technique, it may model complicated connections without making any assumptions about the distribution of the data. The KNN algorithm works as follows:
* 1.Distance calculation: The method determines the distances between each new data point and the input data points in the training set for each new data point metric can be used as the distance measurement, including Manhattan and Euclidean.
* 2.Based on the estimated distances, the algorithm chooses the k nearest data points from the training set. K is a hyperparameter whose value may be selected based on the dataset.
* 3.In order to forecast the value of the new data point, the algorithm first calculates the average or weighted average of the target variable values of the k nearest neighbors. A straightforward and efficient regression approach that can handle non-linear connections and adjust to changes in the data is the KNN regressor. With big datasets, it can be computationally costly and sensitive to the distance measure chosen. Moreover, the KNN regressor makes the assumption that the data distribution is uniform, which could not be the case for all datasets.
* 3.3.4 DECISION TREE REGRESSOR :
* A decision tree is used in the Decision Tree Regressor, a form of regression technique, to forecast the target variable based on a number of input factors. Each node on the decision tree represents a feature or characteristic, and each branch on the decision tree represents a decision rule or condition. The decision tree is constructed by recursively partitioning the dataset depending on the feature that leads to the largest information gain or decrease in the impurity of the target variable. The target variable's impurity is identified using the variance, mean squared error, or any other suitable measure. By navigating the decision tree depending on the values of the input characteristics, we may use it to predict the target variable of a new data point after it has been generated. The decision tree evaluates the value of each node's input characteristic and determines which branch to take in accordance with the decision rule. The target variable's projected value is represented by the leaf node of the tree, where the prediction is made. Over other regression methods, the decision tree regressor offers a number of benefits, including the capacity to handle non-linear connections and complicated decision boundaries, the capacity to manage missing values, and the simplicity of interpretation. Nevertheless, if the tree is too deep or the dataset is too little, the decision tree regressor might potentially experience overfitting. We can employ strategies like pruning, regularization, and ensemble learning to avoid overfitting.
* 3.3.5 RANDOM FOREST REGRESSOR: A regression technique known as the Random Forest Regressor makes predictions by using several decision trees. Because each decision tree in the random forest is trained using a random portion of the training data as well as a random subset of the input characteristics, there is less chance of overfitting and the predictions are more accurate.
* The random forest algorithm works as follows:
* 1. Random subset selection: The algorithm chooses a random subset of the training data and the input characteristics.
* 2. Construction of the decision tree: A decision tree is built using a random subset of characteristics and data. Similar to the decision tree regressor, the decision tree is built by iteratively dividing the data depending on the feature that offers the greatest information gain or impurity reduction.
* 3. Ensemble learning: To generate many decision trees, the first two steps are repeated numerous times. The decision trees' forecasts are then combined using the random forest method to get a final prediction.
* 4. Prediction: The random forest method aggregates the predictions from all of the decision trees in the forest to get a final forecast for each new data point.
* 5. Strong regression algorithms that prevent overfitting, like the Random Forest Regressor, can handle non-linear correlations between the input and target variables. It is frequently used in machine learning applications such as financial forecasting, image analysis, and healthcare prediction. It could struggle to perform well on datasets with few features or training samples.
* 3.4 Evaluation Metrics:
* Evaluation metrics play a critical role in assessing the performance of regression models. They help quantify how well the model predicts the target variable based on the input features. Here are some key evaluation metrics used in regression analysis:
* Mean Squared Error (MSE):
* MSE measures the average squared difference between the actual and predicted values of the target variable. A lower MSE indicates better model performance.
* Root Mean Squared Error (RMSE):
* RMSE is the square root of the MSE. It provides a measure of the average magnitude of the errors in the same units as the target variable.
* R-squared (Coefficient of Determination):
* R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables. It ranges from 0 to 1, with higher values indicating a better fit.

.

* 4. Experimental Setup:
* Model Implementation: The regression models were implemented using Python programming language along with various libraries such as scikit-learn, pandas, and numpy. These libraries provided efficient tools for data manipulation, model training, and evaluation. The implementation was carried out on a standard laptop computer with adequate computational resources.
* Hyperparameter Tuning: The process of hyperparameter tuning involved systematically adjusting the hyperparameters of the regression models to optimize their performance. This was achieved using techniques such as grid search and random search. Grid search involves defining a grid of hyperparameter values and evaluating the model performance for each combination. Random search randomly samples from a distribution of hyperparameter values and evaluates the model performance. The hyperparameters that resulted in the best performance were selected for the final model.
* Cross-Validation: Cross-validation is a technique used to assess the performance and generalization ability of the regression models. In this study, k-fold cross-validation was employed, where the dataset was divided into k subsets or folds. The model was trained on k-1 folds and validated on the remaining fold, repeating this process k times, with each fold used exactly once as the validation data. The average performance across all folds was then calculated to obtain a more reliable estimate of the model's performance. This helped ensure that the model's performance was not overly influenced by the specific training and test data splits.
* 5. Results and Discussion:

● Presentation of Results: The results of the experiments are presented in terms of performance metrics such as Mean Squared Error (MSE), R-squared Score, and others. These metrics provide insights into the predictive accuracy and overall performance of the regression models. Additionally, visualizations such as scatter plots, regression plots, and residual plots may be used to illustrate the relationship between the predicted and actual values, as well as the goodness of fit of the models.

● Interpretation of Findings: The implications of the results are discussed in relation to the research objectives and hypotheses. This involves analyzing how well each regression model performed in predicting the target variable and whether it met the desired criteria for accuracy and reliability. Furthermore, the discussion may delve into the factors that influenced the model's performance, such as the choice of features, data preprocessing techniques, and model complexity.

● Comparison with Previous Studies: The findings are compared with those of previous studies in the field, with a focus on identifying similarities or differences in the results. This comparison helps contextualize the current research within the broader literature and provides insights into the state-of-the-art methods for regression modeling in similar domains. Additionally, any novel contributions or advancements made by the current study are highlighted and discussed in relation to existing knowledge.

* 6. Conclusion:
* This study explored various regression algorithms, including linear regression, ridge regression, KNN regression, decision tree regression, and random forest regression, to predict college admissions based on academic performance and other factors. Through rigorous experimentation and analysis, we observed that each algorithm had its strengths and weaknesses in predicting admission outcomes. Linear regression demonstrated simplicity and interpretability, while ridge regression provided regularization to mitigate overfitting. KNN regression exhibited flexibility in capturing non-linear relationships, while decision tree and random forest regressions offered robustness against complex data patterns.
* Contributions :Our study contributes to the field of machine learning regression by providing a comprehensive evaluation of different algorithms for college admission prediction. By comparing the performance of multiple models, we offer insights into the suitability of each approach and help researchers and practitioners make informed decisions when selecting regression techniques for similar predictive tasks.
* Limitations: Despite our efforts, this study has certain limitations. The performance of the regression models may vary depending on the specific dataset and problem domain. Additionally, we focused primarily on academic performance metrics and did not consider other socio-economic factors that may influence college admissions. Moreover, the evaluation metrics used in this study may not capture all aspects of model performance comprehensively.
* Future Directions:To address the limitations identified in this study, future research could explore the incorporation of additional features, such as extracurricular activities, socioeconomic background, and personal statements, into the predictive models. Furthermore, adopting more sophisticated evaluation metrics, such as area under the ROC curve (AUC) or precision-recall curves, could provide a more nuanced assessment of model performance. Additionally, investigating ensemble techniques that combine multiple regression algorithms may lead to further improvements in predictive accuracy. Overall, there is ample room for future research to enhance the predictive modeling of college admissions and contribute to more equitable and efficient admission processes.
* References: [1] Pachauri, S., & Singh, S. P. (2019). Predictive modeling for student admission using machine learning algorithms. Journal of Education and Practice, 10(1), 78-89.
* [2] Bhowmik, S. K., & Tiwari, R. K. (2020). Application of machine learning techniques for student admission prediction. Journal of Education and Practice, 11(4), 40-49.
* [3] Singh, D. (2018). Predicting student admission using machine learning techniques. International Journal of Advanced Research in Computer Science, 9(2), 62-67.
* [4] Khare, R., & Sharma, V. (2021). Admission prediction in higher education using machine learning. In Proceedings of the International Conference on Advanced Computing and Intelligent Engineering (pp. 53-60). Springer.
* [5] Gupta, A., & Gupta, R. (2018). Admission prediction in higher education institutes using machine learning algorithms. International Journal of Computer Science and Mobile Computing, 7(1), 143-151.
* [6] Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists.
* [7] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.
* [8] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective.
* [9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.
* [10] Bishop, C. M. (2006). Pattern Recognition and Machine Learning.
* [11] Directorate of Technical Education [Online]. Available: http://www.dte.org.in/fe2011/StaticPages/Default.aspx
* [12] Kaggle. (n.d.). Student Admission Prediction Dataset. Retrieved from [Dataset-link](file:///C:\Users\vlanj\Downloads\Dataset-link)
* Appendices (if necessary): Detailed Algorithm Descriptions:
* Linear Regression: Explanation of the theory and implementation of linear regression.
* Ridge Regression: Description of the ridge regression technique and its application.
* KNN Regressor: Overview of the K-nearest neighbors regression algorithm and its parameters.
* Decision Tree Regressor: Explanation of decision tree regression and its key components.
* Random Forest Regressor: Description of random forest regression and its advantages.
* Additional Experimental Results:
* Detailed Performance Metrics: Tables showing metrics such as MSE, RMSE, and R-squared for each model.
* Model Evaluation Visualizations: Visual representations of model evaluation metrics for comparison.
* Code Snippets:
* Model Implementation Code: Code snippets demonstrating the implementation of each regression model.
* Hyperparameter Tuning Code: Code examples for tuning hyperparameters and optimizing model performance.
* Cross-Validation Code: Examples of cross-validation techniques used to ensure robustness of the results.