* Enhancing Percentile prediction
* of Engineering Colleges: Regression Models
* Vaibhav Nivrutti Lanjewar
* **Department of Information Technology, Shri Guru Gobind Singhji**
* **Institute of Engineering and Technology, (SGGSIET), Nanded**
* [2021bit023@sggs.ac.in](file:///C:\Users\vlanj\Downloads\2021bit023@sggs.ac.in)

**Abstract:**

**Securing admission to an engineering program is akin to navigating a complex maze, with myriad factors influencing one's chances of success. To streamline this process and provide students with valuable insights, this study introduces a novel approach employing decision trees to predict admission probabilities accurately. By harnessing the power of decision trees, which function as intelligent flowcharts, this method aims to empower students to make informed decisions while fostering fairness and simplicity in the admissions process.**

**Accurate predictions of acceptance rates play a pivotal role in helping students assess their chances and aiding colleges in selecting the most suitable candidates. By leveraging decision trees, our method has demonstrated remarkable effectiveness in providing precise estimates of admission probabilities.**

**Accessibility is key in ensuring the widespread adoption and usability of predictive models. To this end, we have developed a user-friendly tool that simplifies the input process, requiring only basic information such as gender, category, and preferred college. This tool provides instant estimates of admission probabilities, making it equally accessible for students and admissions personnel.**

**Ensuring the reliability and accuracy of predictive models is essential. Our decision tree model underwent rigorous testing and validation procedures to confirm its efficacy. Comparative analysis against alternative methods revealed that our model outperformed others, boasting minimal errors and high accuracy. Notably, the mean squared error of 3.546854e-01 and an impressive R-squared score of 0.999 underscore the robustness of our approach.**

**In our pursuit of refinement, we actively sought feedback from admissions experts. Their insights and suggestions were invaluable in fine-tuning our methodology, further enhancing the accuracy and applicability of our predictive model.**

**Our study represents a significant leap forward in simplifying and enhancing the college admissions process. By leveraging decision trees, we have developed a powerful tool that provides students and colleges alike with accurate and accessible estimates of admission probabilities. This innovation not only empowers students to make informed decisions but also facilitates fair and efficient admissions practices. As we continue to refine and improve our methodology, we remain committed to advancing the field of college admissions through innovative and user-centric approaches.**

**Keywords:** Regression Model, Ridge regressor, Random Forest and K nearest regressor, admission to engineering colleges.

**1. Introduction:**

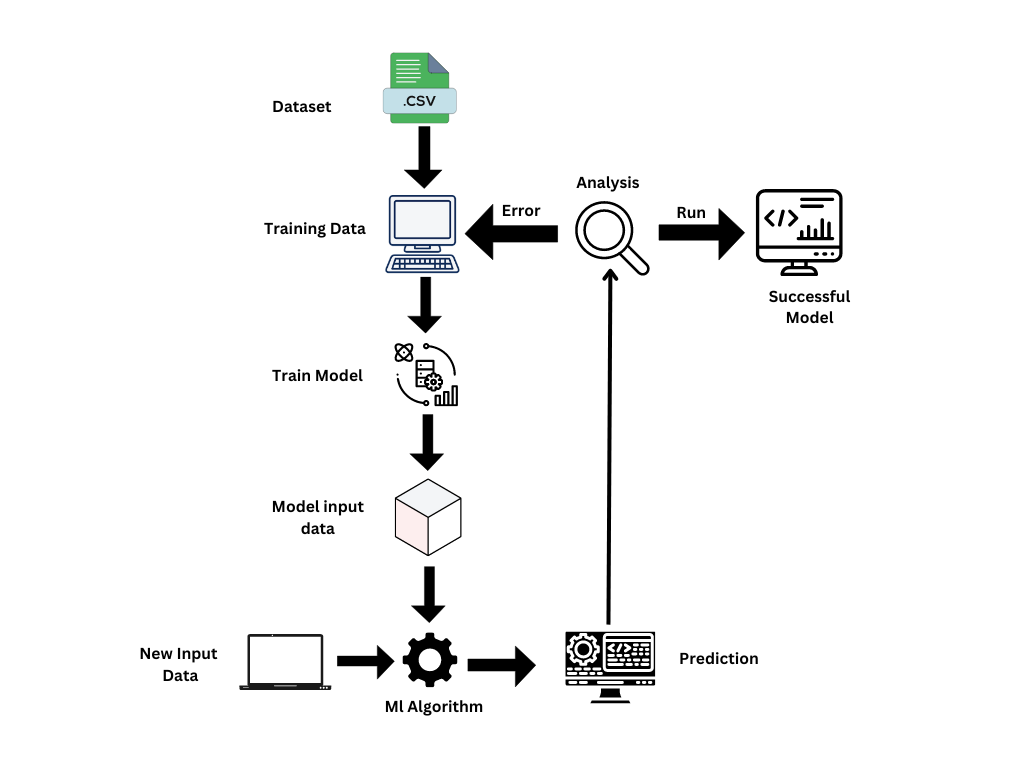
In today's competitive business environment, it is important for students who want to be successful in their careers to attend prestigious institutions. This is especially crucial for aspiring engineers, where the demand for technical skills is high. However, the complexity of the admission process poses a significant challenge for students, particularly in states like Maharashtra, where numerous engineering colleges and branches offer a wide array of options.

With over 1.5 million seats spread across 200+ colleges and 35+ branches, sorting through the available choices can be daunting. Students must carefully consider their academic studies and interests when compiling a list of potential universities. However, navigating relationships and competition often leads to suboptimal decisions.

To address this challenge, a computer-based approach to improving college choice is invaluable. By leveraging academic data and user preferences, our tool aims to predict the best university for each student. Factors such as course selection, location, and exclusion criteria are taken into account to generate a curated list of institutions that align with students' needs and aspirations.

Our tool empowers students to make informed decisions and increases their chances of admission to prestigious universities. By streamlining the admissions process, we aim to mitigate the risks associated with university and college selection, making the transition to higher education smoother and more manageable.

**1.1 SYSTEM ARCHITECTURE**

* 
* fig-1: System Architecture of the prediction model

**2. Literature Review:**

Machine learning regression has emerged as a powerful tool in predictive modeling, offering insights into complex data and facilitating decision-making across various domains. Numerous studies have explored different algorithms, methodologies, and techniques within machine learning regression.

Horizontal regression, ridge regression, decision tree regression, KNN regression, and random forest regression stand out as the primary techniques employed in predictive modeling. These techniques model relationships ranging from linear equations to complex interactions. For instance, ridge regression addresses multicollinearity issues by introducing a penalty term. Decision tree regression partitions the feature space into regions and predicts responses based on the average of training events within each region. KNN regression estimates response deviation from the mean of its nearest neighbors' values. Random forest regression integrates multiple decision trees to enhance prediction accuracy and robustness.

Despite the advancements, there remain gaps and flaws in the existing data. Notably, there is a lack of a comprehensive model that incorporates the myriad factors influencing college admissions, including academics, course preferences, and geographic considerations. Existing models often focus solely on forecasting product competition characteristics, overlooking the broader challenges of admission preferences and limitations. Moreover, existing systems may lack transparency or require specialized expertise, limiting student access.

To address these gaps, a process is underway to combine advanced machine learning techniques and utilize multiple predictive models. The aim is to provide personalized recommendations, empowering students to make informed decisions about their education. Through this approach, the model seeks to bridge the existing gaps and offer tailored guidance to students navigating the complexities of college admissions.

2.1 Existing Research Paper

Existing research on machine learning regression has explored various algorithms and methodologies. The following regression algorithms have been studied:

Linear Regression

K-Nearest Neighbors (KNN) Regressor

Decision Tree Regressor

Random Forest Regressor

Identification of Gaps:

Evaluation Metrics: The provided data includes accuracy as a performance metric for regression algorithms. However, additional metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared could provide a more comprehensive evaluation.

Analysis Across Test Sizes: The data presents model performance across different test sizes, but there is a lack of detailed analysis on how each algorithm's accuracy varies with varying test sizes. Understanding the impact of test size on model performance is crucial for assessing generalization capabilities.

Algorithm Selection Criteria: The literature review does not discuss the criteria for selecting a specific regression algorithm for a given problem. Factors such as dataset characteristics, linearity of relationships, interpretability, and computational efficiency should be considered in algorithm selection.

Trade-off Between Interpretability and Performance: While some algorithms may offer interpretability, their accuracy might be lower compared to more complex models. Investigating the trade-off between model interpretability and performance is essential, especially in domains where both aspects are critical.

RELATED WORK:

*Paper 1*- ADMISSION PREDICTION IN ENGINEERING AND TECHNOLOGY COLLEGES by Anuja Sawarkar, Rakhi Gupta, Shreyas Tayade, Rutvik Tale[13].

Model summary: In this research paper, the focus is on algorithm selection and model training for predicting students' likelihood of admission to engineering and technology universities. The study employs regression techniques, namely Linear Regression, KNN Regressor, Decision Tree Regressor, and Random Forest Regressor, to forecast admission probabilities based on students' academic performance and AIEEE rank.

The data is divided into training and testing sets, with varying test sizes of 0.1, 0.2, 0.3, and 0.4. The choice of regression algorithm depends on the nature of the data and the problem statement. In this context, where multiple independent variables (10th Marks, 12th Marks, 12th Division, and AIEEE rank) are used to predict a single dependent variable (College Rank), the performance of each algorithm is evaluated across different test sizes.

Results indicate that the Decision Tree Regressor consistently outperforms other algorithms across various test sizes, exhibiting high accuracy in predicting admission probabilities. Conversely, the Linear Regression model shows relatively lower performance compared to other algorithms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Test size=0.1 | Test size=0.2 | Test size=0.3 | Test size=0.4 |
| Linear Regression | 12.40 | 31.10 | 27.76 | 29.85 |
| KNN Regressor | 90.85 | 90.83 | 91.26 | 77.42 |
| Decision Tree Regressor | 90.37 | 95.86 | 97.53 | 84.08 |
| Random Forest Regressor | 77.16 | 85.46 | 83.60 | 84.08 |

Fig-3: Summary of model performance

*Paper 2-*  College Admission Prediction using Ensemble Machine Learning Models Vandit Manish Jain1, Rihaan Satia (2021)

In this research paper, they aim to develop a predictive model to assist students in selecting suitable universities across various domains, such as MS (international), MTech (India), and MBA (India and International). We train a machine learning model using a comprehensive dataset comprising student profiles and university details, including admission outcomes. Utilizing Ensemble Machine Learning algorithms, we evaluate their performance using key performance indicators (KPIs) to identify the best-performing model. The dependent variable, indicating the likelihood of admission to a university, ranges from 0 to 1, representing the predicted probability of successful acceptance.

These accuracy scores underscore the effectiveness of the models in predicting admission outcomes based on student profiles and university details. Furthermore, our objective includes developing a portal to simplify the university selection process for students by providing personalized recommendations tailored to their profiles and preferences.

|  |  |
| --- | --- |
| Model | Accuracy |
| Linear Regression | 0.8212 |
| Neural Networks | 0.7447 |
| Decision Trees | 0.6588 |
| Random Forest | 0.7909 |

*Paper* 3- Prediction for University Admission Using Machine Learning

Summary: In this research paper, the

|  |  |
| --- | --- |
| Algorithm | Accuracy |
| Linear Regression | 0.79 |
| Ridge Regression | 0.78 |
| Random Forest | 0.77 |
| KNN | 0.72 |

**3. Methodology:**

Data Cleaning and Feature Engineering:

The data sourced from "mht\_cet2.csv" underwent thorough cleaning to address missing values, discrepancies, and inconsistencies. Missing data were handled using appropriate techniques, such as mean, median, or mode imputation, or by removing rows with missing values altogether to preserve data integrity. Additionally, the results were processed using methods like truncation, separation, or statistical measures to mitigate their impact on the model. The cleaned datasets were made compatible for further analysis.

Dataset:

In general, data for a machine learning project can be gathered from a variety of sources, including surveys, online scraping, public datasets, and collaborations with businesses. In the instance of Admission Prediction in Engineering & Technology Schools, we discovered the information we could utilize from Kaggle. The information may have been gathered through previous admission records of the institutions, student transcripts, or through surveys given to applicants. To prepare the data for analysis and modeling, it was pre-processed and cleaned to remove any discrepancies or inaccuracies. By eliminating missing values, scaling the numerical features, and transforming category characteristics into numerical ones, we cleaned and pre-processed the data.

A screenshot of a computer

Description automatically generated

Fig-2: Unprocessed Dataset

After preprocessing the dataset by various techniques and algorithms :A screenshot of a white screen

Description automatically generated

Fig-2: Processed Dataset

Feature Engineering:

Feature engineering is a crucial step aimed at ensuring that all features contribute equally to the training sample, thereby avoiding bias due to small differences. In this process, the model references data to predict additional functions, which may involve creating correlation coefficients, multivariate specifications, or aggregating data from different sources to provide better input for regression models. Features such as 'Secondary seat type', 'score\_type', 'college\_name', and 'branch' are encoded using techniques such as one-hot encoding or label encoding to convert them into a numerical format suitable for model training.

Box-Plot for Outlier:

A box plot, also known as a box-and-whisker plot, is a graphical representation of the distribution of a dataset. It provides a visual summary of the central tendency, dispersion, and skewness of the data, including any potential outliers.

The formula for Identifying Outliers:

In a box plot, outliers are typically identified using the interquartile range (IQR) method. The IQR is calculated as the difference between the third quartile (Q3) and the first quartile (Q1) of the dataset.

Calculate the Interquartile Range (IQR): IQR=Q3−Q1

Define the Lower and Upper Bounds:

Lower Bound: Q1−1.5×IQR

Upper Bound: Q3+1.5×IQR

Box-Plot:

A box plot visually represents the following statistical measures:

Minimum: the smallest data point within the dataset.

First Quartile (Q1): the 25th percentile, representing the lower quartile of the data.

Median: the middle value of the dataset, also known as the 50th percentile.

Third Quartile (Q3): the 75th percentile, representing the upper quartile of the data.

Maximum: the largest data point within the dataset.

Whiskers: lines extending from the box to the minimum and maximum values that are within 1.5 times the IQR from the first and third quartiles, respectively.

Outliers: individual data points that fall beyond the whiskers.

**Outlier detection for our model**:

* Upper Bound: 144.6930989
* Lower Bound: -14.380961
* Original Data Frame shape: (104345, 11)
* Data Frame shape after removing outliers: (104345, 11)

It appears that *no outliers* were identified and removed from the dataset, as the Data Frame shape remains the same before and after the outlier removal process. This suggests that all data points fall within the acceptable range defined by the upper and lower bounds. Therefore, there were no data points considered as outliers according to the defined criteria.

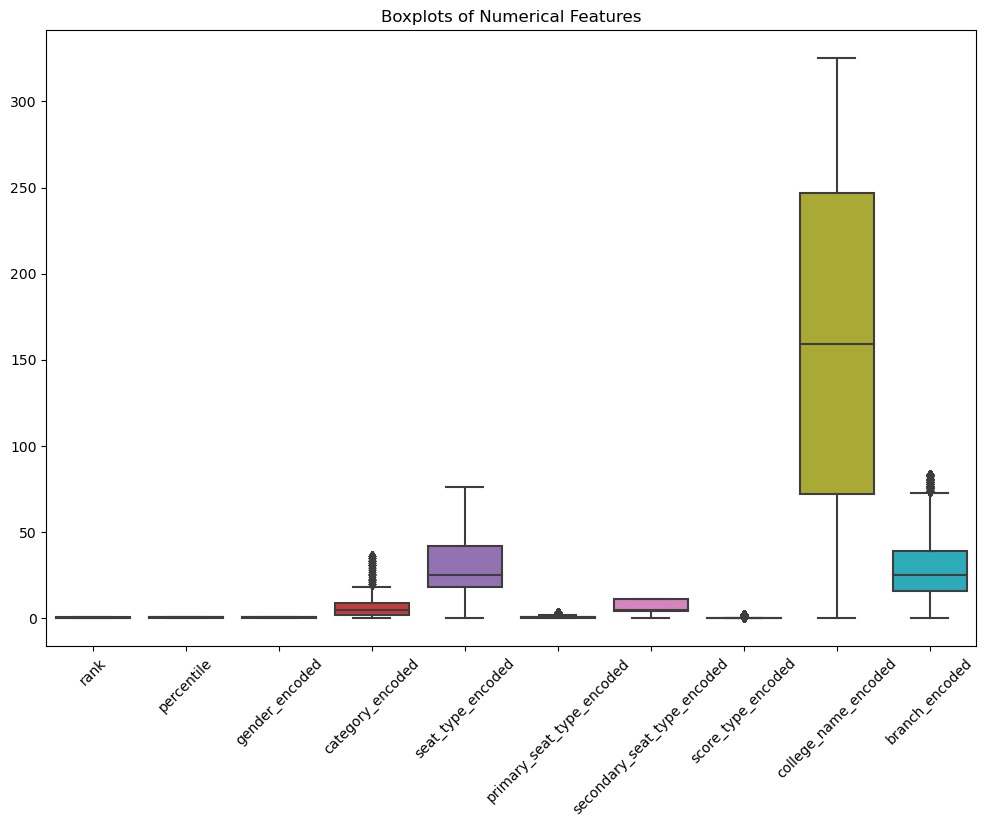


Fig-4: Box-plot

**Correlation:**

Corr(X,Y) = Cov[X,Y] / ( StdDev(X) ∙ StdDev(Y) )

Correlation analysis is a fundamental statistical technique used to explore the relationship between variables in a dataset. It helps us understand how changes in one variable are associated with changes in another variable. In our context, correlation analysis will provide insights into how different features in our dataset relate to each other, which can aid in feature selection and model building.

The correlation coefficient, typically denoted as "r," quantifies the strength and direction of the linear relationship between two variables. It ranges from -1 to 1:

A correlation coefficient close to 1 indicates a strong positive correlation, meaning that as one variable increases, the other variable tends to increase as well.

A correlation coefficient close to -1 indicates a strong negative correlation, meaning that as one variable increases, the other variable tends to decrease.

A correlation coefficient close to 0 suggests a weak or no linear relationship between the variables.

Pearson's correlation coefficient is commonly used to measure the linear correlation between two continuous variables. It is calculated as the covariance of the two variables divided by the product of their standard deviations.

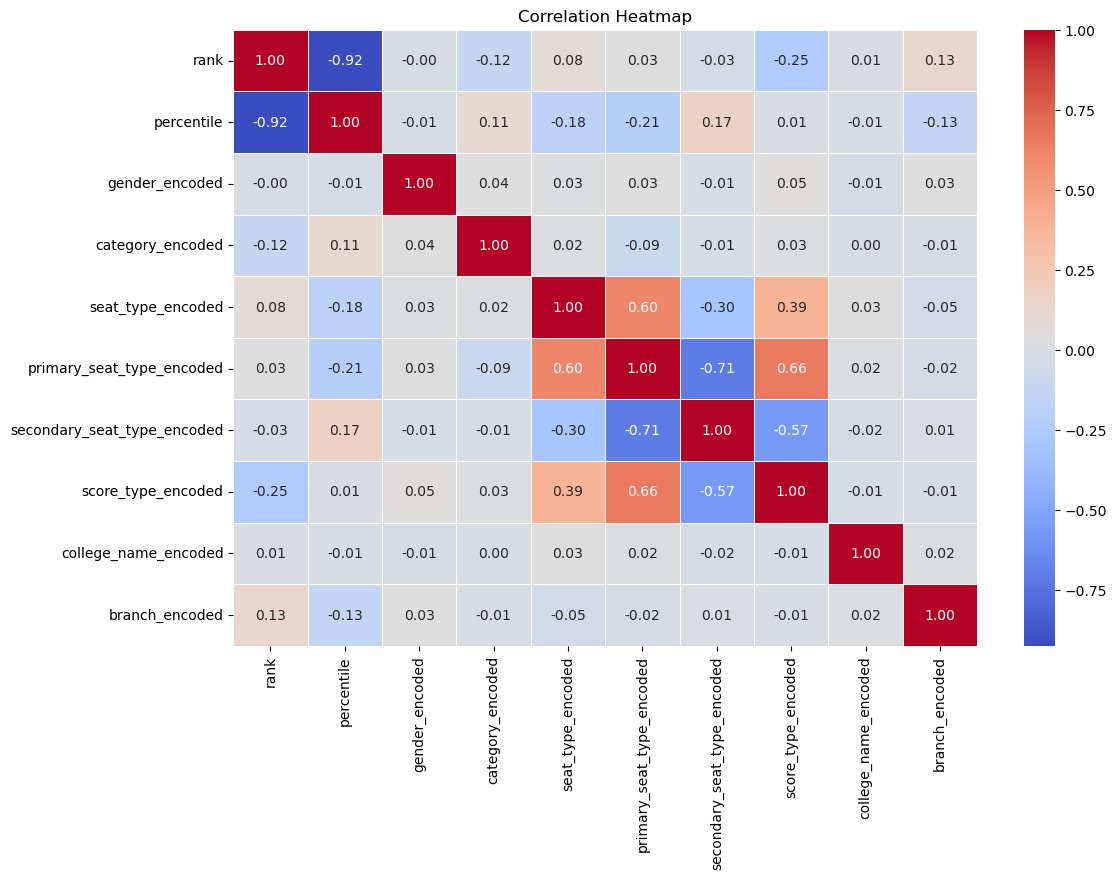


Fig-5:Correlation

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 a test size: of 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.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:

3.3.1 LINEAR REGRESSION:

It is a form of regression technique called linear regression that 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 y = β0 + β1x1 + β2x2 + ... + βnxn

represents the line of best fit.

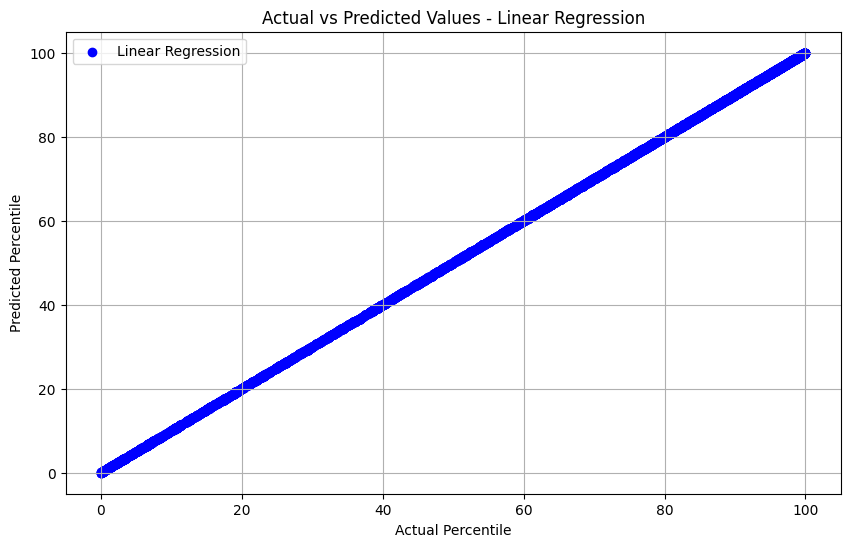
****

Fig-6: Linear Regression

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.

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.

A graph with red lines

Description automatically generated

Fig-7: Ridge Regression

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**.

Formulae:

Manhattan Distance = | x 1 − x 2 | + | y 1 − y 2 |

Euclidean Distance = ( x 2 − x 1 )2 + ( y 2 − y 1 ) 2

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.

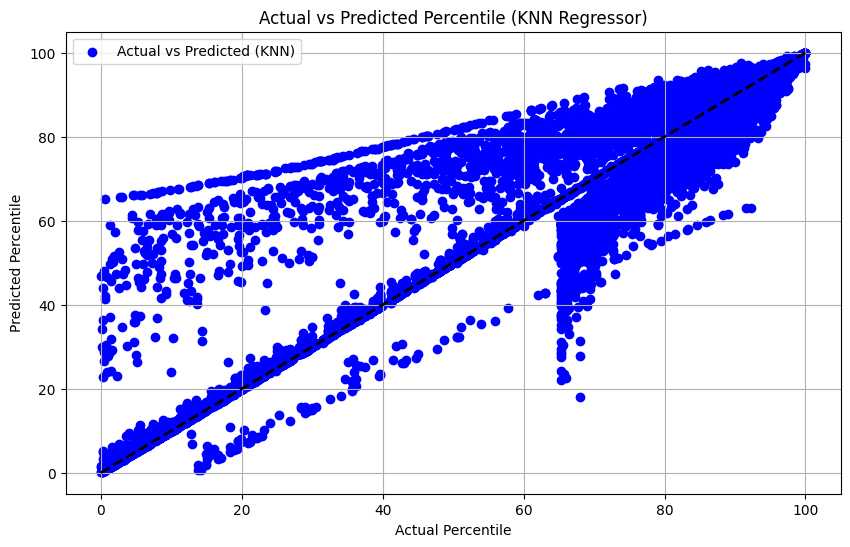


Fig-8: KNN regressor

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.

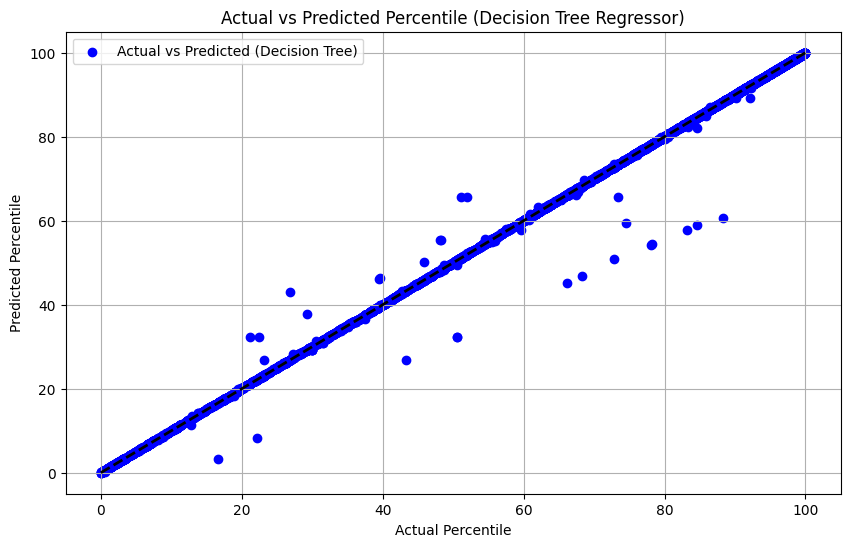


Fig-9: the Decision Tree Regressor

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.

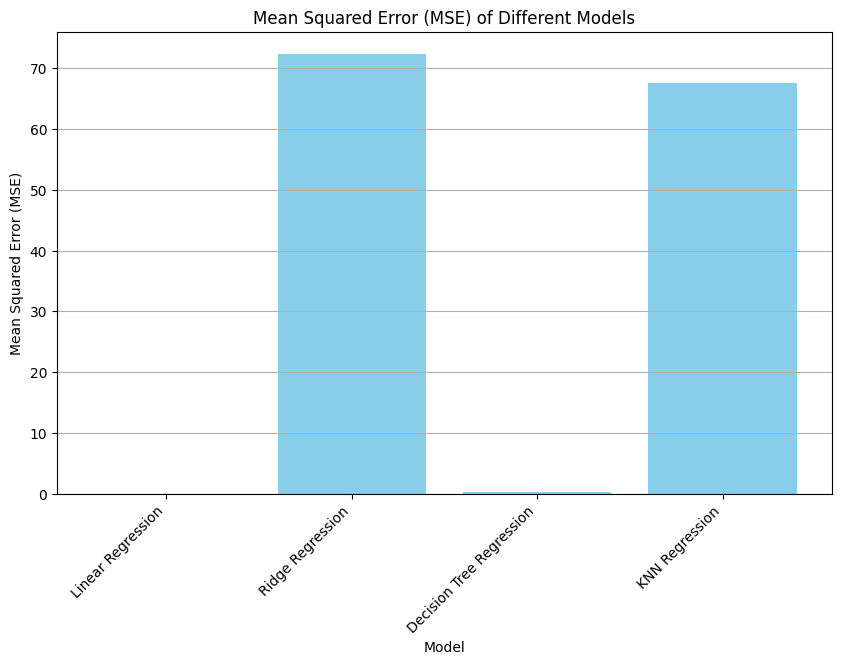


Fig-10: MSE for Different Models

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.

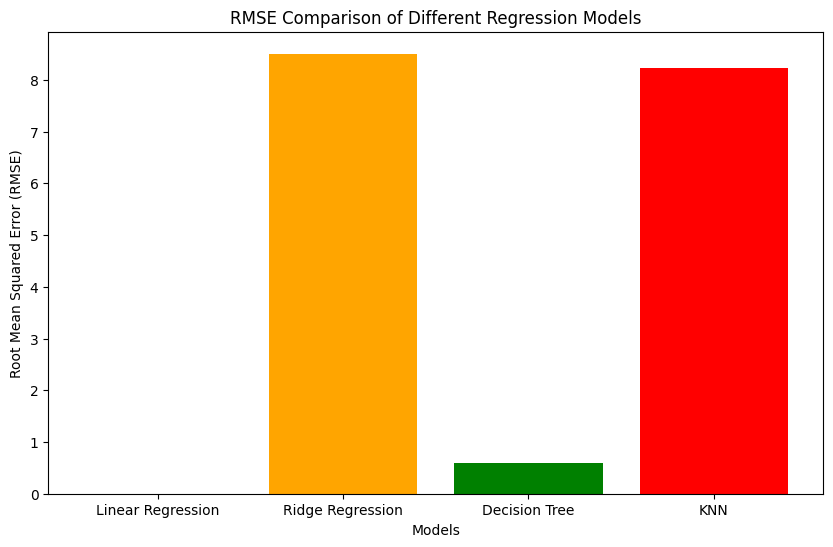


Fig-11 **:**Rmse for Different Models

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.

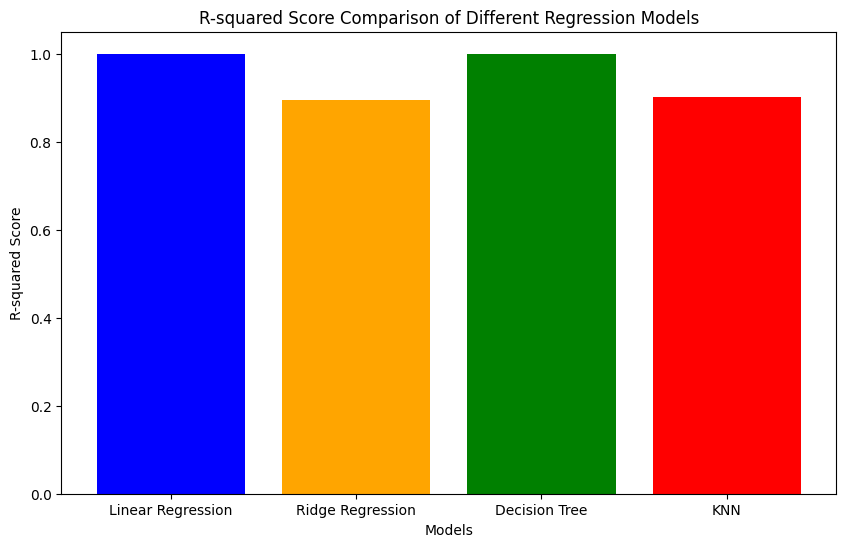


Fig-12 **:** R-squaredfor Different Models

|  |  |  |  |
| --- | --- | --- | --- |
| Model No. | Model | Mean Squared Error(MSE) | R-Squared Score |
| 1 | Linear Regression | 7.161513e-26 | 1.000000 |
| 2 | Ridge Regression | 7.233226e+01 | 0.893709 |
| 3 | Decision Tree Regression | 3.546854e-01 | 0.999479 |
| 4 | KNN Regression | 6.756597e+01 | 0.900713 |

**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.

Cross-Validation and hyperparameter tuning for model-

Ridge-Regression:

Alpha: The regularization strength parameter. Increasing alpha shrinks the coefficients towards zero, which can help reduce overfitting by penalizing large coefficients. It controls the amount of regularization applied to the model. A higher alpha value leads to a more regularized model, which may result in better generalization performance on unseen data.

Ridge regression tends to be less sensitive to outliers compared to ordinary least squares regression. The regularization term in Ridge regression helps to smooth out the effect of outliers by penalizing large coefficients, reducing their influence on the model.

Mean Squared Error: 72.33 225696348437

R-squared Score: 0.8937087833641754

Cross-Validation

RMSE Scores:[7.19978942,7.51038144,11.09167267,7.30778736,7.42296424]

Mean Cross-Validation RMSE: 8.106519025249565

Decision Tree Regression:

Max Depth: The maximum depth of the decision tree. Increasing max depth allows the tree to grow deeper, potentially capturing more complex relationships in the data. However, deeper trees may lead to overfitting, especially with noisy or sparse data.

Min Samples Split: The minimum number of samples required to split an internal node. Increasing this parameter can prevent the model from creating nodes with very few samples, which helps to control overfitting.

Min Samples Leaf: The minimum number of samples required to be at a leaf node. Similar to min samples split, increasing this parameter can help prevent overfitting by requiring a minimum number of samples in each leaf.

Decision trees can be sensitive to outliers, especially when using metrics like mean squared error for splitting nodes. Outliers can disproportionately influence splits by causing the tree to partition the feature space in a way that overfits the training data. However, by controlling parameters like max depth and min samples split, decision trees can mitigate the impact of outliers to some extent.

Mean Squared Error (Decision Tree): 0.3546853726098193

R-squared Score (Decision Tree): 0.9994787949199946

Best parameters: {'max\_depth': None, 'min\_samples\_leaf': 1, 'min\_samples\_split': 5}

K nearest Regression:

The number of neighbors to consider. Increasing this parameter can lead to a smoother decision boundary and potentially improve generalization. However, too few neighbors can result in overfitting, while too many neighbors can lead to underfitting.

Weights: Specifies the weight function used in prediction. "Uniform" weights all points in each neighborhood equally, while "distance" weights points by the inverse of their distance. Using "distance" can give higher weight to closer neighbors, potentially improving prediction accuracy.

Algorithm: The algorithm used to compute nearest neighbors. Options include "auto", "*ball\_tree*", "kd\_tree", and "brute". Each algorithm has its own computational trade-offs, but they generally produce similar results.

KNN is sensitive to outliers because it relies on distance metrics to determine nearest neighbors. Outliers can disproportionately affect distance calculations and influence predictions. However, using "distance" weighting can help mitigate this issue by giving less weight to outliers that are far from the query point.

Mean Squared Error (KNN): 67.56596501986324

R-squared Score (KNN): 0.9007127811764484

Best parameters: {'algorithm': 'ball\_tree', 'n\_neighbors': 9, 'weights': 'distance'}

**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.

Summary of Model Performance :

|  |  |  |
| --- | --- | --- |
| **Model No** | **Model Name** | **R-squared Score** |
| 1 | Linear Regression | 1.000000 |
| 2 | Ridge Regression | 0.893709 |
| 3 | Decision Tree | 0.999479 |
| 4 | KNN | 0.900713 |

* **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.

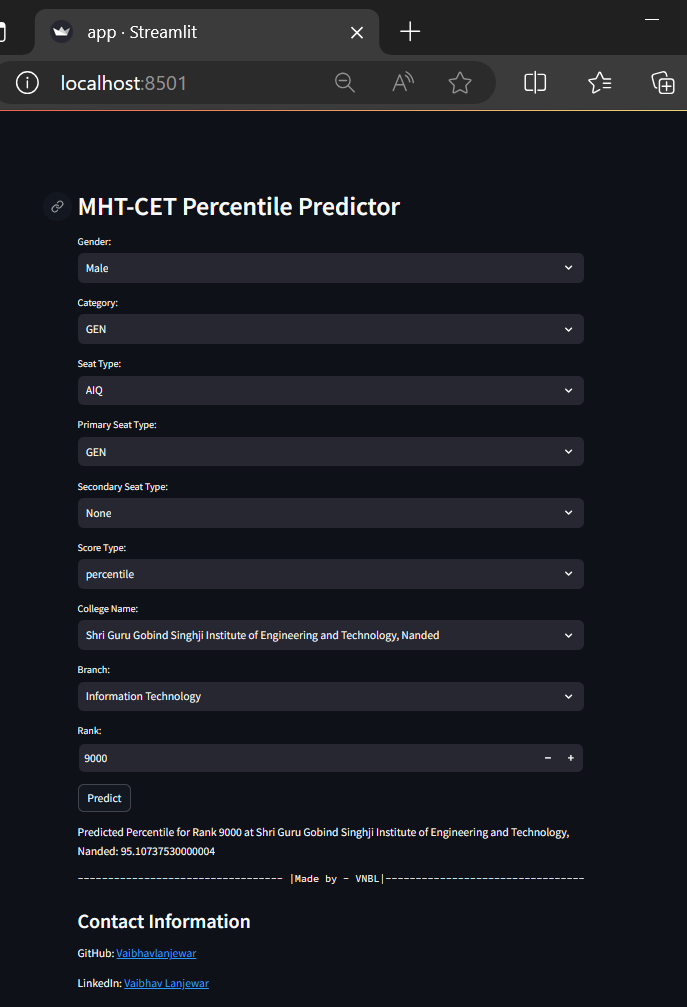
Contributions : Our study contributes to the field of machine learning regression by providing a comprehensive evaluation of different algorithms for college admission prediction. By comparing the performance of multiple models, we offer insights into the suitability of each approach and help researchers 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.

.

**8. User Friendly- Interface and Deployment :**

Model is deployed using streamlit ,which provides functionality to use as Website



* **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)
* [13] For **Literature review , model paper for Regression model** –
* <https://www.irjet.net/archives/V10/i3/IRJET-V10I395.pdf>
* **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.