IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Data-Driven Insights for Medical College Allotment: Evaluating ML Models for College Prediction.

Chandra Harika Shatdharshanam, Mahitha Gudipati, Naman Kumar Muktha, Teja Karthik Thangudu

Department of Computer Science Engineering- Artificial Intelligence & Machine Learning, CMR College of Engineering & Technology, 501401, Hyderabad, India

Abstract: The segregation or the classification of the medical colleges in India based on the NEET rank of the students is a very vital task that can significantly impact the quality of the healthcare sector of the country. This research paper provides an analysis of multiple factors influencing the ranking of the medical colleges in India and it proposes a classification framework based on these factors.

This research utilizes a dataset containing information about medical colleges, including factors like score, gender, category, locality, etc. Various machine learning such as random forests, decision trees have been implemented on the dataset, but when compared, the best accuracy attained was through decision trees. The findings of this research paper can be used by various educational institutions and students to improve the quality of medical education in India.

Index Terms- Machine learning techniques, Predictive model, Classification framework, Decision trees, Random forests, Categorical data, Discrete variables, Model Interpretation, Overfitting, Feature Importance, Linear models, Computational resources, Class imbalance.

1. Introduction

Medical education in India going through some significant technological advancements, which inevitably leads to evolving medical practices and an increase in demand for the quality of healthcare services. The very first step in increasing or merely evaluating the quality of medical education begins at the initial stages where the students who are inevitably going to be future professionals choose their very own medical college based on their rank and also where the parents and policymakers make decisions regarding the education and career choices. The segregation or classification and ranking of medical colleges in India are typically based on various factors including quality of the faculty, student performance, overall academic excellence, facilities, infrastructure. These rankings play a vital role in shaping the reputation of the medical colleges and they influence the student enrolment. Despite the enormous amount of data available on the internet and the importance of rankings, there is a lack of comprehensive study that systematically organizes and analyses the factors that are influencing the ranks. Our research paper aims at filling this gap by conducting a detailed analysis of the influence of the various factors on the ranks of the students and in turn their selection in the college, this paper also aims at developing a classification framework using machine learning techniques. By leveraging machine learning techniques and algorithms, this research seeks to identify the most significant

factors contributing to the selection of students in various medical colleges and develop a predictive model that can easily suggest the medical colleges based on the student's rank and the factors about why and how the ranks are classified. This classification framework can provide valuable insights for stakeholders in their healthcare and education sectors, enabling them to make an informed decision to improve the quality of medical education and healthcare services in India. using a dataset predominantly composed of categorical and discrete variables. Given the nature of our data, it was imperative to select algorithms capable of effectively handling such characteristics. Our first choice for modelling was decision trees. Decision trees offer a natural fit for datasets with discrete variables due to their inherent ability to handle categorical data directly. This characteristic made decision trees an attractive option for our research, as it allowed us to build a model that could effectively capture the nuances of our **dataset.** Moreover, the transparency of decision trees in decision-making processes facilitated model interpretation, enabling us to understand and communicate the underlying logic driving predictions. Despite their simplicity, decision trees often yield competitive accuracy, striking a balance between bias and variance. This aspect was particularly advantageous for our research, where achieving accurate predictions was crucial for meeting our objectives. Additionally, decision trees are well-suited for multi-class classification tasks, making them a practical choice for our analysis.

2. Literature Review

In a paper that is concerned with classifying the research papers in 3 classes i.e. science, social science, and business, many ML algorithms such as KNN, SVM, Decision trees, Decision tree method was proved to be one of the most intuitive machine learning methods amongst the non-parametric supervised machine learning algorithms that can be used for both classification and regression test, and each leaf node (terminal node) holds a class label[1]. The learning algorithm behind decision tree is an inductive approach to learn knowledge on classification by splitting the source datasets into subsets based on an attribute value test [2]. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has the same value of the target variable, or when splitting no longer adds value to the predictions. Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop. There are four types of decision tree algorithms, namely Iterative Dichotomiser (ID3), Classification and Regression Trees (CART), Chi-square, and Reduction in Variance.ID3 decision tree algorithm, uses Information Gain to decide the splitting points.

Decision trees are a popular and widely-studied approach for classification in machine learning and data mining. Rokach and Maimon (2005) provide a comprehensive survey of methods for constructing decision tree classifiers in a top-down manner. They present a unified algorithmic framework and describe various splitting criteria and pruning methodologies used in decision tree induction.[3]

The authors discuss several key aspects of decision trees, including univariate and multivariate splitting criteria, handling of missing values, and pruning techniques. Common univariate splitting criteria covered include information gain, gain ratio, and Gini index. The paper also reviews pruning methods like cost-complexity pruning, reduced error pruning, and pessimistic pruning that aim to improve generalization by reducing tree size. Additionally, the authors summarize popular decision tree algorithms such as ID3, C4.5, CART, and CHAID, highlighting their key characteristics and differences.

The survey examines extensions to classical decision trees, including oblivious decision trees, fuzzy decision trees, and incremental induction methods. It also discusses techniques for handling large datasets and addresses the advantages and limitations of decision trees as a classification approach. Overall, this paper provides a thorough overview of decision tree methods, serving as a valuable resource for researchers and practitioners working with this important family of machine learning algorithms.

3. Methods

Our first choice for modelling was decision trees. Decision trees offer a natural fit for datasets with discrete variables due to their inherent ability to handle categorical data directly. This characteristic made decision trees an attractive option for our research, as it allowed us to build a model that could effectively capture the nuances of our dataset. Moreover, the transparency of decision trees in decision-making processes facilitated model interpretation, enabling us to understand and communicate the underlying logic driving predictions.

Despite their simplicity, decision trees often yield competitive accuracy, striking a balance between bias and variance. This aspect was particularly advantageous for our research, where achieving accurate predictions was crucial for meeting our objectives. Additionally, decision trees are well-suited for multi-class classification tasks, making them a practical choice for our analysis. As an extension of decision trees, we also considered employing random forests in our modelling approach.[4] One of the primary advantages of random forests is their ability to mitigate the issue of overfitting, which is a common concern with decision trees. By aggregating predictions from multiple trees, random forests offer a robust solution for combatting overfitting, thereby enhancing the generalizability of our model. Furthermore, the ensemble nature of random forests typically leads to improved prediction accuracy compared to individual decision trees. This enhancement in accuracy was desirable for our research, where precise predictions were essential for informing decision-making[5]. Additionally, random forests provide reliable estimates of feature importance, allowing us to identify the most influential variables driving predictions[6]. This insight proved valuable for understanding the underlying mechanisms of our model and identifying areas for further investigation. In contrast to decision trees and random forests, other algorithms such as linear models and neural networks were not considered suitable for our research objectives. Linear models, while effective in certain contexts, have limitations in handling categorical data effectively, which comprised a significant portion of our dataset. Similarly, neural networks, while powerful in their ability to capture complex patterns, were deemed unnecessary given the performance achieved by decision trees and random forests. In conclusion, the selection of decision trees and random forests as our primary modelling algorithms proved instrumental in addressing the challenges posed by our dataset. Their ability to handle discrete data, maintain interpretability, mitigate overfitting, and deliver competitive accuracy enabled us to effectively meet our research objectives. By leveraging these algorithms, we were able to develop a robust model capable of making accurate predictions and providing valuable insights into the factors influencing the outcome variable. In our dataset, we observed class imbalance, which could potentially bias our model towards the majority class [7]. To address this, we employed a Random Over Sampler. This technique works by duplicating instances from the minority class to achieve class balance.

BEFORE USING RANDOM SAMPLER: The above graph represents the class distribution before oversampling. The varying heights of the bars indicate the presence of class imbalance in our dataset.[8]

4. Steps/Procedure:

1. Importing Dependencies

Start by importing the necessary libraries. These will be used for data manipulation, visualization, and model building.[9]

a. NumPy - (Numerical Python) is a foundational library for numerical and scientific computing in Python. It provides support for arrays, matrices, and many mathematical functions to operate on these data structures.

b. Pandas - Pandas is a powerful data manipulation and analysis library for Python. It provides data structures and functions needed to manipulate structured data seamlessly. It is used for manipulation of data frames, data manipulation, and time series analysis.[10]

c. Matplotlib - Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides a variety of plotting functions and a high level of control over the visual appearance of plots. It is used for plotting, customization, and integration.

d. Seaborn - Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It is used for visualizations, statistical plotting themes, and customization. [11]

Together, these libraries form a powerful toolkit for data analysis and machine learning in Python.

2. Reading Files [12]

a. Import the data from all the phases - Since the data is split into multiple files, read each file into a Pandas DataFrame.

b. Concatenate all the files into a single file - Combine all the DataFrames into one for ease of model training.

3. Data Preprocessing

Data preprocessing is the process of transforming raw data into a format that is suitable for analysis and modeling. This step is crucial in the data analysis and machine learning pipeline as it ensures the quality and integrity of the data, which in turn affects the performance and accuracy of models. Preprocessing steps typically include handling missing values, encoding categorical variables, sampling, and more.

Dealing with the missing values - Missing values in a dataset can pose several issues:

a. Bias: Missing values can introduce bias, leading to inaccurate conclusions and predictions.

b. Reduced Data Quality: They can affect the quality and reliability of the analysis.

c. Algorithm Compatibility: Many machine learning algorithms cannot handle missing values directly and require a complete dataset to function properly.

Label Encoding - Label encoding is the process of converting categorical data into numerical data by assigning a unique integer to each category. [13]

a. One-Hot Encoding: One-hot encoding converts categorical variables into a series of binary columns, each representing a unique category. Each column contains a 1 (true) or 0 (false).

b. Categorical Encoding: Categorical encoding is a broader term that includes various techniques to convert categorical data into numerical data.

KBinsDiscretization - KBinsDiscretizer is a transformer in the Scikit-learn library that discretizes continuous features into k bins. Discretization, also known as binning, transforms continuous data into discrete intervals or bins. This can be useful for various purposes, such as simplifying models, handling outliers, and preparing data for algorithms that require discrete input.[14]

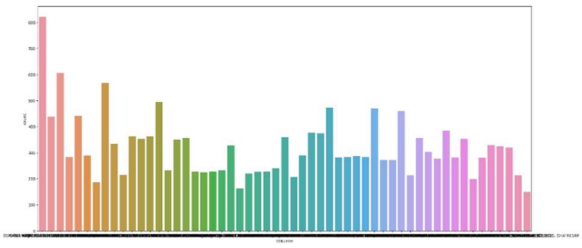
We use Discretization for the following reasons:

a. Simplifying Models: Binning can reduce the complexity of models by converting continuous variables into categorical ones.

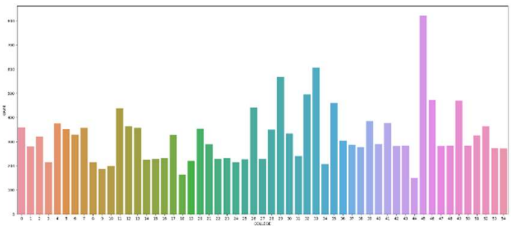
b. Handling Outliers: Binning can help mitigate the influence of outliers by placing them into the same bin as fewer extreme values.

c. Feature Engineering: Binning can create meaningful features for certain types of models, such as decision trees.[15]

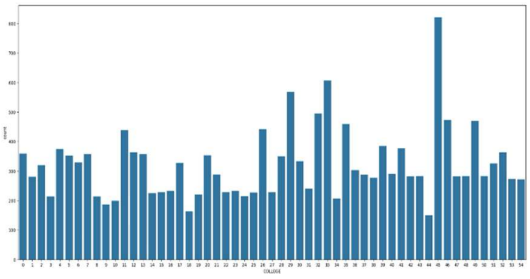
d. Sampling - Sampling is a critical technique in data analysis and machine learning for managing large datasets, improving computational efficiency, and ensuring statistical validity. By selecting a representative subset of data, we can perform quicker analyses, develop models faster, and make more robust inferences about the overall population [8]



Data before KBinsDiscretization, represented using Matplotlib and Seaborn

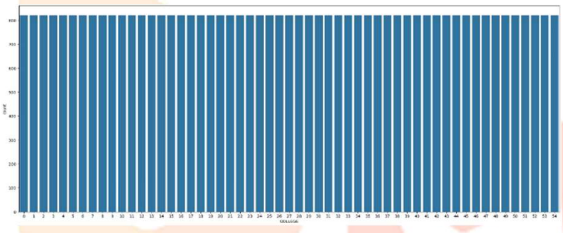


Data after KBinsDiscretization, represented using Matplotlib and Seaborn



Data before sampling

BEFORE USING RANDOM SAMPLER: The above graph represents the class distribution before oversampling. The varying heights of the bars indicate the presence of class imbalance in our dataset.



Data after sampling

AFTER USING RANDOM SAMPLER: The above graph represents the class distribution after applying Random Over Sampler. As can be seen, the classes are now balanced, with each class having an equal representation in the dataset.

4. Split Training and Testing data[16]

Splitting data into training and testing sets is a fundamental practice in machine learning and data science. This procedure helps ensure that a model generalizes well to new, unseen data. Here are the key reasons why we split data into training and testing sets:

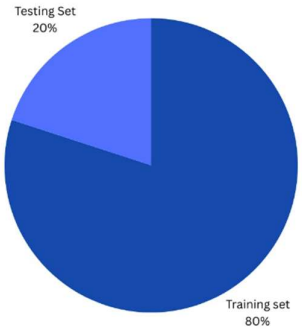
a. Assessing Model Performance

b. Model Validation

c. Hyperparameter Tuning

d. Preventing Data Leakage

By appropriately splitting the data, we can develop robust models that perform well on new, unseen data.



5. Models Training

Random Forest - Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting. It is widely used for both classification and regression tasks.[6]

a. Method

b. Bagging

c. Random Feature Selection

Decision Trees Classifier - Decision Trees are a type of supervised learning algorithm used for both classification and regression tasks. They split the data into subsets based on the value of input features, forming a tree-like structure.

a. Nodes and Leaves

b. Splitting Criteria

c. Recursive Partitioning

KNearestNeighbours - K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm used for classification and regression tasks. It predicts the class of a data point based on the classes of its k-nearest neighbours [17]

a. Distance Metric (KNN Model-Based Approach in Classification)

b. Majority Vote

c. Averaging

Classification Metrics

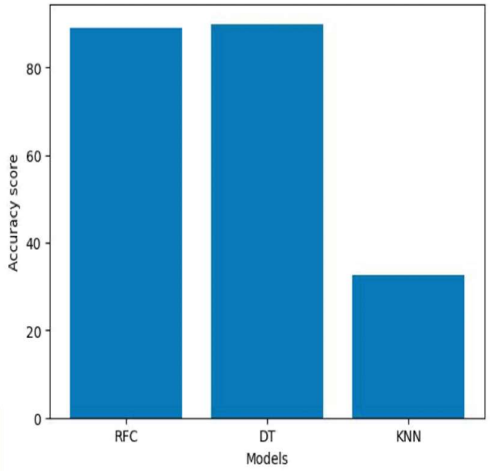
a. Accuracy - Accuracy is a commonly used metric to evaluate the performance of a classification model. It is defined as the ratio of the number of correct predictions to the total number of predictions made. Accuracy is a straightforward and intuitive measure of a model's performance

Accuracy = Number of Correct Predictions / Total Number of Predictions

b. Confusion Matrix - A confusion matrix is a table used to evaluate the performance of a classification model. It provides a comprehensive view of the model's predictions compared to the actual outcomes, showing the counts of true positives, true negatives, false positives, and false negatives.

In conclusion, the selection of decision trees and random forests as our primary modelling algorithms proved instrumental in addressing the challenges posed by our dataset. Their ability to handle discrete data, maintain interpretability, mitigate overfitting, and deliver competitive accuracy enabled us to effectively meet our research objectives. By leveraging these algorithms, we were able to develop a robust model capable of making accurate predictions and providing valuable insights into the factors influencing the outcome variable.

In our dataset, we observed class imbalance, which could potentially bias our model towards the majority class. To address this, we employed a Random Over Sampler. This technique works by duplicating instances from the minority class to achieve class balance.



Comparing accuracies of different models

5. Conclusion

The present study aims to develop a classification model for various medical colleges in the states of Andhra Pradesh, and Telangana, India, by employing machine learning techniques, specifically the decision tree algorithm. The motivation behind this research stemmed from the need to provide a systematic and data-driven approach to categorizing medical educational institutions, thereby facilitating informed decision-making processes for stakeholders, including students, parents, and policymakers. The model was trained on features such as rank, locality, gender, category, and economic strength. This approach was undertaken to address the challenges faced by stakeholders in navigating the complex higher education landscape and making informed choices aligning with their academic goals.

The implementation of the decision tree algorithm involved a rigorous data preprocessing stage, where the relevant features were carefully selected, cleaned, and transformed to ensure optimal performance. The algorithm's ability to handle both numerical and categorical data made it an ideal choice for this classification task. Through recursive partitioning, the decision tree effectively identified the most significant factors influencing the categorization of medical colleges, enabling the construction of a robust and interpretable model. The model's performance was evaluated using appropriate metrics to ensure its reliability and validity.

Overall, this study contributes to the growing body of knowledge in the field of educational data mining and highlights the potential of machine learning techniques in streamlining decision-making processes within the higher education sector.

REFERENCES

[1] Osisanwo, F. Y., et al. "Supervised machine learning algorithms: classification and comparison." International Journal of Computer Trends and Technology (IJCTT) 48.3 (2017): 128-138.

[2] S. Yousefi, H. Karimipour, and F. Derakhshan, “Data Aggregation Mechanisms on the Internet of Things: A Systematic Literature Review,” Internet of Things (Netherlands), vol. 15, Sep. 2021, doi: 10.1016/j.iot.2021.100427.

[3] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," IJCSI Int. J. Comput. Sci. Issues, vol. 9, no. 5, pp. 272–278, 2012.

[4] B. Mahesh, "Machine Learning Algorithms - A Review," Int. J. Sci. Res., vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/art20203995.

[5] X. Ying, "An Overview of Overfitting and its Solutions," J. Phys. Conf. Ser., vol. 1168, no. 2, 2019, doi: 10.1088/1742-6596/1168/2/022022.

[6] A. Liaw and M. Wiener, "The R Journal: Classification and regression by randomForest," R J., vol. 2, no. 3, pp. 18–22, 2002, [Online]. Available: http://www.stat.berkeley.edu/

[7] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0192-5.

[8] S. Noor and O. Tajik, "Simple Random Sampling," Sampl. Popul. Methods Appl. Fourth Ed., vol. 1, no. November, pp. 43–81, 2011, doi: 10.1002/9780470374597.ch3.

[9] P. Gupta and A. Bagchi, "Introduction to NumPy," pp. 127–159, 2024, doi: 10.1007/978-3-031-43725-0_4.

[10] A. Sapre and S. Vartak, "Scientific Computing and Data Analysis using NumPy and Pandas," Int. Res. J. Eng. Technol., pp. 1334–1346, 2020.

[11] A. Oberoi and R. Chauhan, "Visualizing data using Matplotlib and Seaborn libraries in Python for data science," Int. J. Sci. Res. Publ., vol. 9, no. 3, p. p8733, 2019, doi: 10.29322/ijsrp.9.03.2019.p8733.

[12] J. R. Schmidt, "CSVDataMerge: A Simple and Free Program for Concatenating Experimental Data Files," J. Open Res. Softw., vol. 9, 2021, doi: 10.5334/JORS.368.

[13] Y. Kementchedjhieva and I. Chalkidis, "An Exploration of Encoder-Decoder Approaches to Multi-Label Classification for Legal and Biomedical Text," Proc. Annu. Meet. Assoc. Comput. Linguist., no. d, pp. 5828–5843, 2023, doi: 10.18653/v1/2023.findings-acl.360.

[14] S. Cycles, "Chapter 9 Chapter 9," Cycle, vol. 1897, no. Figure 1, pp. 44–45, 1989, doi: 10.1007/0-387-25465-X.

[15] A. Zien, N. Krämer, S. Sonnenburg, and G. Rätsch, "The feature importance ranking measure," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5782 LNAI, no. PART 2, pp. 694–709, 2009, doi: 10.1007/978-3-642-04174-7_45.

[16] G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes, "Experimental Evaluation of Train and Test Split Strategies in Link Prediction," Stud. Comput. Intell., vol. 944, pp. 79–91, 2021, doi: 10.1007/978-3-030-65351-4_7.

[17] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2888, no. November 2012, pp. 986–996, 2003, doi: 10.1007/978-3-540-39964-3_62