International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

Envisioning Student's Performance Over Placements into IT Sector

P Meher Sunitha, Narayan Soni, N Likhitha, A Durga Prasad,

Dr. Venkata Ramana Mancha.

Abstract: **Anticipating student placements is an essential task for educational institutions, as it can help identify students who may need additional support and resources to succeed in their careers. This project aims to predict student placement performance based on academic performance, testing their fundamentals, standardised test scores, and many more. To accomplish this, we collected data from a large sample of students who had appeared for placement training & attended the skill ability test and later analysed their performance concerning these factors. Furthermore, we shall use this data to develop a machine-learning model to predict student performance on placements. The model trains on the collected data and will be able to predict student performance accurately. This model can help educational institutions [1] or organisations to identify students who may be at risk of poor performance on placements and provide them with the necessary support and resources to improve their skills and efficiency.**

I. INTRODUCTION

P. Meher Sunitha, Research Student, Dept. of CSE, GITAM(Deemed to be University) Visakhapatnam, mail id: 121910310022@gitam.in.

Narayan Soni, Research Student, Dept. of CSE, GITAM(Deemed to be University) Visakhapatnam, mail id: 121910310004@gitam.in.

N. Likhitha, Research Student, Dept. of CSE, GITAM(Deemed to be University) Visakhapatnam, mail id: 121910310040@gitam.in

A Durga Prasad, Research Student, Dept. of CSE, GITAM(Deemed to be University) Visakhapatnam, mail id: 121910310015@gitam.in

Dr. Venkata Ramana Mancha, Assistant Professor, Dept. of CSE, GITAM(Deemed to be University) Visakhapatnam, mail id: vmancha@gitam.edu.

IN today's competitive job market, it is crucial for educational institutions to ensure that their students are well-equipped for their future careers. One of many ways to approach is by providing students with hands-on experience through placements or internships [2]. However, it is essential to ensure that students perform well in their placements, as poor performance can adversely impact their career prospects.

To address this challenge, this study aims to predict student performance in placements by examining several parameters, such as academic performance, standardized test scores, and fundamentals & coding skills. By understanding the correlation between these factors and student performance in placements, educational institutions can identify students who may require additional support and resources to succeed in their placements.

The findings of this study can significantly benefit educational institutions by helping them tailor their support and resources to meet the needs of their students more effectively. As a result, this can lead to better outcomes for students, resulting in a more competitive workforce.

It is worth noting that the success of this study largely depends on the accuracy of the predictive model developed. Therefore, it is crucial to consider a diverse range of data sources to ensure that the model is not biased and provides accurate predictions.

In conclusion, this study has the potential to be a game-changer for educational institutions by enabling them to predict student performance in placements accurately. By providing additional support and resources to those who require it, educational institutions can ensure that their students have the necessary skills to succeed in their future careers, leading to better job prospects and a more competitive.

II. RELATED WORKS

Several studies have explored the factors that influence student performance in placements or internships.

One such study by Zhao and Wang (2020) [3] analysed the impact of academic performance and career development on student performance in internships. The results of the study indicated that academic performance had a positive effect on student performance in internships, while career development had a negative effect.

Similarly, a study by Wong et al. (2018) [4] found that students who had higher academic performance and participated in more extracurricular activities had better performance in internships. Additionally, the study also highlighted the importance of communication skills, time management, and adaptability in predicting student performance in internships.

In terms of technical skills, a study by Chua et al. (2018) [5] found that students who had better coding skills had higher chances of securing internships and performing well in them. Similarly, a study by Tsai et al. (2019) [6] found that programming ability was a significant predictor of student performance in software development internships. The use of machine learning models has also been explored in predicting student performance in placements. For instance, a study by Kulkarni et al. (2019) [7] used machine learning algorithms to predict student performance in placements based on factors such as academic performance, technical skills, and personality traits.

Overall, the literature suggests that academic performance, extracurricular activities, technical skills such as coding, and

communication skills are significant predictors of student performance in placements or internships and help educational institutions in providing appropriate support and resources to their students to succeed in their careers.

III. Materials & Methods

In accordance with this journal, this chapter in particular, refers to the technologies and techniques or even a software , an application utilised in order to develop this placement prediction system. The dataset is actually acquired from a report generated by GCGC (Gitam Career Guidance Center).This dataset contains records of Batch: 2019–2023 student's data on placement training examination testing their cognitive and intellectual ability to get into IT sector.

The report considered the following parameters and predicted whether a student's performance falls under which category: (1) good to go (2)Need Practice (3)Low performance-requires training.

Reg_No	Unique Student ID
Marks_tenth	The percentage obtained as per their Secondary School Certificate
Marks_twelfth	The percentage obtained as per their Higher School Certificate
Graduation_CGPA	The Cumulative Grade Point Average obtained till the latest semester
Aptitude	The score testing their mathematical & Numerical reasoning in the exam.
Computer_Fundamental	The score testing their computational & networking knowledge in the exam
Coding	The score testing their logical, mental reasoning and programming ability in the exam.

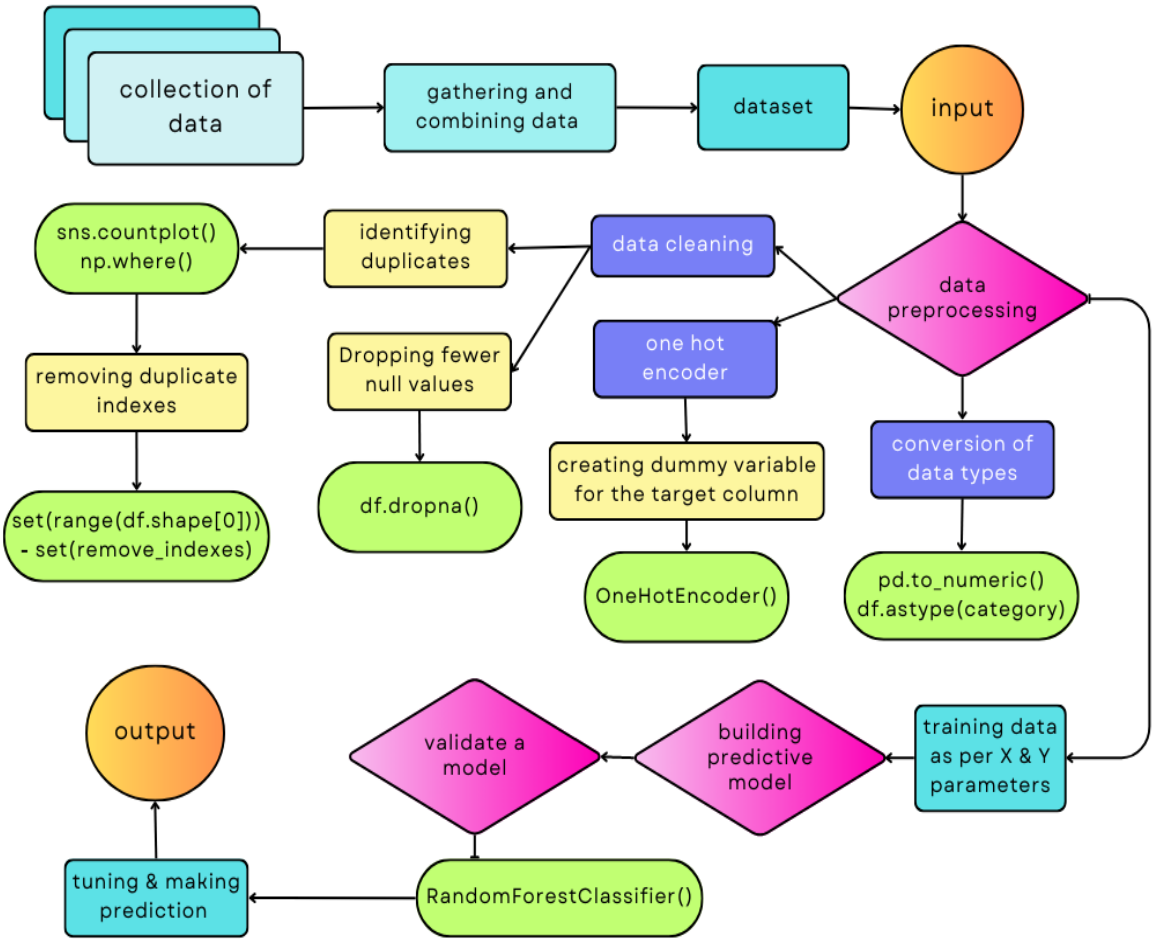

Fig 3.1: *Architecture flow diagram explaining the complete process of the model.*

Parameter	Description

IV. IMPLEMENTATION

The implementation of this project involves collecting and processing data, building, and training predictive models, and developing a web application for visualizing and presenting the results. This is achieved using a range of technologies and tools such as Python, Jupyter Notebook, scikit-learn, and Streamlit and is conveniently briefed in 3 steps: Coding, validating and Deployment.

- Coding comprises of steps like performing EDA [8] i.e., Data collection, Data cleaning, Data Preparation, Data Visualisation.
- Validating involves building predictive model, training the data, testing with different ML models, and validating the model.
- Deployment process constitute of integrating the backend code into the front-end part using pickle.

A. Data Collection & Importing:

Data collection and importing is a critical aspect of exploratory data analysis. It involves the process of locating and loading data into a system. Reliable and high-quality data can be sourced from various public sites or purchased from private organizations. Some dependable sites for data collection include Kaggle, GitHub, the Machine Learning Repository etc.

For our project, we have utilized a specific dataset is acquired from a report generated by GCGC(GITAM Career Guidance Center).Additionally, we import the data as per our requirements and for easy access to Python libraries. Once the data is imported, we read the CSV file into the kernel and explore other data characteristics using functions and methods.

B. Data Cleaning:

Data cleaning is a crucial process in exploratory data analysis where we remove irrelevant variables and values from a dataset and address any irregularities. Irregularities or anomalies can significantly affect the accuracy of the results, and hence it is essential to address them. To clean the data, we take various steps such as removing missing values, outliers, and redundant rows/columns, and re-indexing and reformatting the data.

One of the primary tasks in data cleaning is to identify the missing values in each column and the proportion of missing values they contribute. We can use different functions and methods to perform these actions, which help us determine the extent of data cleaning required. By thoroughly cleaning the data, we can improve the accuracy and reliability of our results, enabling us to make better-informed decisions.

In this project, the procedure of data cleaning involved dropping fewer null values, identifying duplicates from the data with the help of seaborn counterplot and NumPy library in order to remove the indexes.

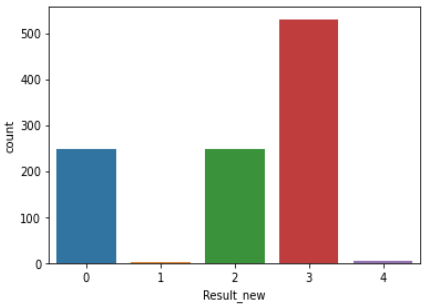

Fig 4.1 *identifying duplicate values using count plot.*

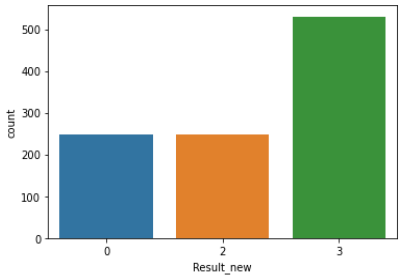

Fig 4.2 *after removing duplicate values using NumPy.*

C. Data Preparation:

Data preparation encompasses gathering, structuring, and arranging data in a way that is appropriate for use in data visualization, business intelligence (BI), and analytics applications. This procedure entails converting raw data into a format that is convenient to access and utilize, making analysis and decision-making more efficient. It involves various tasks, including data cleansing, normalization, transformation, and enrichment, and guarantees that the data is precise, consistent, and of excellent quality. Data preparation is a crucial phase in data analysis since it ensures that the data is properly organized and ready for analysis, thereby enhancing overall operational efficiency.

- **Converting Data Type:**

In machine learning, the data used to train a model comes in different formats, such as numerical, categorical, or text data. Different algorithms and models require data to be in specific formats or structures to work correctly. For example, a decision tree algorithm may only accept categorical data, while a linear regression model may only work with numerical data.

Therefore, changing or converting the datatypes in a machine learning model involves modifying the data's format or structure to meet the model's requirements. This process ensures that the model can process the input data correctly and make accurate predictions or classifications.

There are different methods to change or convert datatypes, depending on the type of data and the model used. For instance, numerical data can be converted into categorical data by dividing it

into ranges or bins. Text data can be transformed into numerical data using techniques such as tokenization or vectorization.

Overall, changing or converting datatypes is an essential step in machine learning data pre-processing to ensure that the model can handle the input data correctly and generate accurate results.

```
In [6]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1045 entries, 0 to 1044
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Reg_No                1045 non-null   int64
 1   Branch                1045 non-null   object
 2   Marks_tenth           1045 non-null   object
 3   Marks_twelfth         1045 non-null   object
 4   Graduation_CGPA       1045 non-null   object
 5   Aptitude              1045 non-null   int64
 6   Computer_Fundamental  1045 non-null   int64
 7   Coding                1045 non-null   int64
 8   Over all %            1045 non-null   object
 9   Result                1045 non-null   object
dtypes: int64(4), object(6)
memory usage: 81.8+ KB
```

Fig 4.4 *After data preprocessing*

```
In [9]: #rechecking the datatypes after conversion
        df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1045 entries, 0 to 1044
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Reg_No                1045 non-null   int64
 1   Branch                1045 non-null   category
 2   Marks_tenth           1039 non-null   float64
 3   Marks_twelfth         1039 non-null   float64
 4   Graduation_CGPA       1039 non-null   float64
 5   Aptitude              1045 non-null   int64
 6   Computer_Fundamental  1045 non-null   int64
 7   Coding                1045 non-null   int64
 8   Over all %            1040 non-null   float64
 9   Result                1045 non-null   category
dtypes: category(2), float64(4), int64(4)
memory usage: 68.0 KB
```

- **One Hot Encoder:**

One hot encoding [9] is a procedure utilised to transform categorical data into a numerical format that can be easily processed by machine learning models. It works by creating a binary representation of each category, where each category is represented by a vector of 0s and 1s.

Here is an example to illustrate one hot encoding: let us say we have a dataset of animals, and one of the features is their type, which can be "dog," "cat," or "bird." Since this feature is categorical, we need to convert it into a numerical format that a model can understand.

To do this, we can use one hot encoding to create a binary representation of each animal's type. For instance, we can represent:

- "dog" as [1,0,0],
- "cat" as [0,1,0],
- and "bird" as [0,0,1].

Each vector contains a 1 in the position corresponding to the animal's type and 0s in the other positions.

By using one hot encoding, we can now represent the categorical data as numerical data, and models can process it more easily. For example, if we want to predict whether an animal can fly or not based on its type and other features, the model can use the one hot encoded vector of the animal's type as input to make its prediction.

Here as we observe, the target column i.e., the result parameter is of categorical data type and cannot be taken into consideration while building a predictive model, as majority of the models require int or float values to consider. Thus, we create a dummy variable [Result_new] using one hot encoder which assigns the values periodically.

Furthermore, it can noticed that:

- The value “0” is assigned to – Good to go.
- The value “2” is assigned to – Need Practice
- The value “3” is assigned to – Low performance-need training.

```
Percentage of "Good to go: " 24.22178988326848 %
Percentage of "Need practice: " 24.22178988326848 %
Percentage of "Low performance-need training:  51.55642023346303 %
```

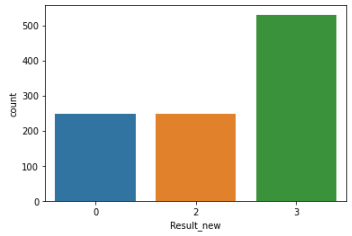

Fig 4.5 *count plot for displaying the new dummy variable.*

D. Data Visualization:

- **Co Relation Matrix:**

A correlation matrix [10] is a table that shows the relationship between variables in a dataset. It is used in data analysis and machine learning to identify patterns and dependencies between variables, and to determine which variables are most relevant to a given problem.

A correlation matrix consists of a square table where the rows and columns represent the variables in the dataset, and each cell in the table represents the correlation between two variables. The correlation can range from -1 to 1, where -1 indicates a -ve correlation (when one variable increases, the other decreases) and 1 indicates a +ve correlation (when one variable increases, the other also increases). A correlation of 0 indicates no correlation between the variables.

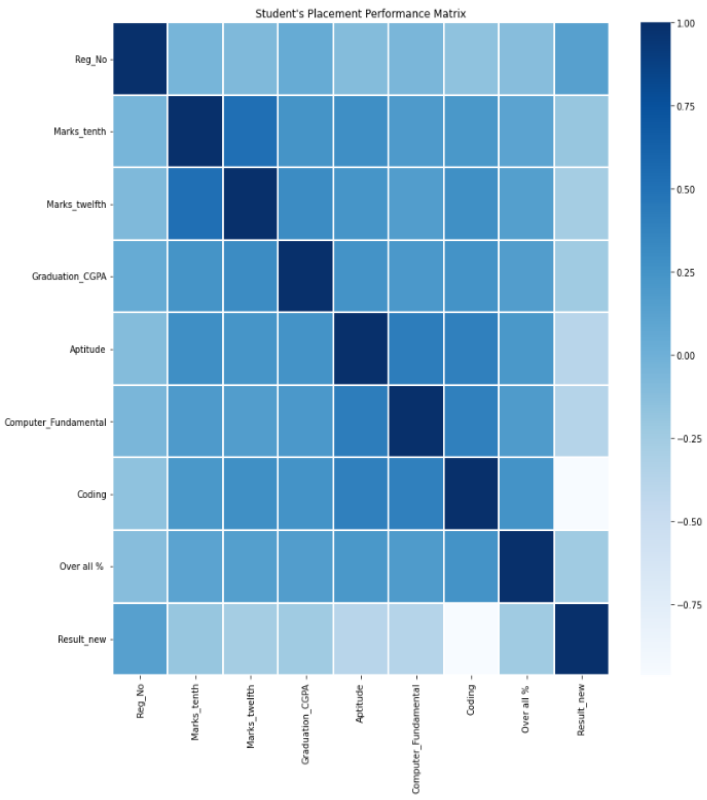

Fig 4.6 *Correlation matrix*

- **Seaborn Pair plot**

A pair plot [11] consists of a matrix of scatterplots, where each scatterplot shows the relationship between two variables in the dataset. The diagonal of the matrix typically shows a histogram or a density plot of each variable, allowing you to visualize the distribution of the data for each variable. They are useful for identifying patterns and relationships in the data that may not be apparent from individual scatterplots or histograms. They can help you to identify variables that are strongly correlated or have nonlinear relationships, as well as outliers or other anomalies in the data.

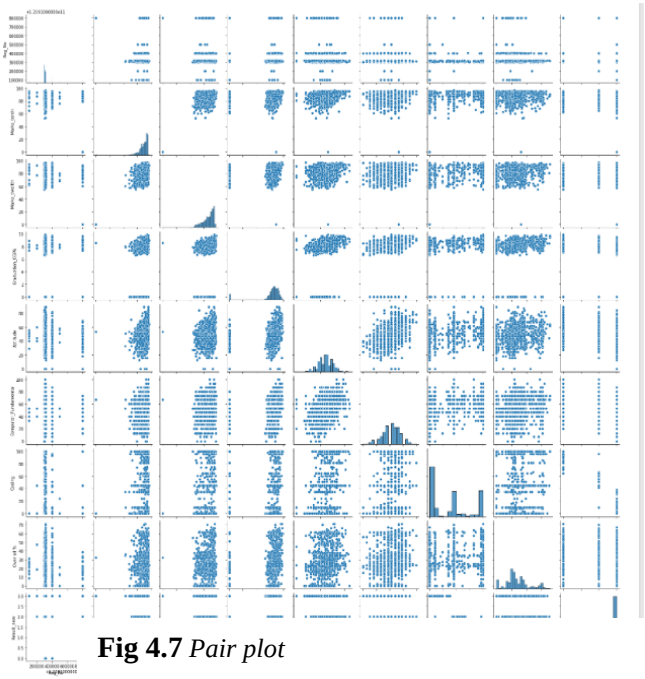

Fig 4.7 *Pair plot*

V. Results & Discussion

The results of this journal indicate that a predictive model based on machine learning techniques can effectively predict student performance on placements using factors such as academic performance, standardized test scores, and fundamentals. The model's accuracy was evaluated using several metrics, including precision, recall, and accuracy[13], and the performance of different algorithms and models was compared.

The model was trained, tested & validated using the above classifiers:

- **Linear models:** Support Vector Machine, Linear Regression, Naïve Bayes.
- **Non-Linear Model:** K-Neighbors, Random Forest

The result of our analysis revealed that linear regression and random forest classifier models [14] yielded better results than other models. The accuracy scores for these models were 0.99 and 0.96, respectively, while the accuracy scores for other models were lower, with the lowest being 0.52. Based on these results, we proceeded with the random forest classifier model for our final model as it has accuracy of 0.99.

The

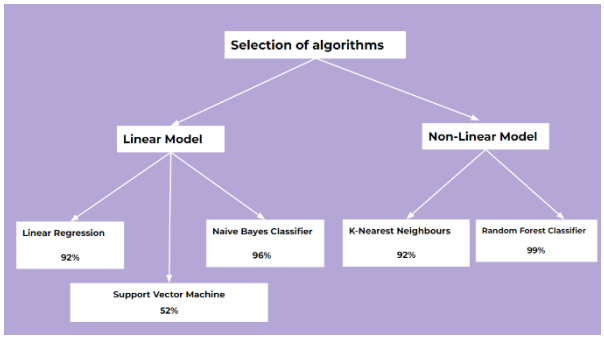

Fig 5.1 *Accuracies of different classifier algorithms*

Discussion constituted the success of our model highlights & the importance of experimenting with different machine learning models to identify the best fit for a particular problem. By comparing different algorithms and techniques, we were able to identify the models that yielded the best results for our specific problem. It is important to note, however, that accuracy is not the only feature that should be used to evaluate the performance of a predictive model. Additionally, the study highlights the importance of considering multiple metrics and preventing overfitting to ensure that the final model is reliable and effective in making.

Here are the sample testcases detected from the model:

Student ID	10th grade score	12th grade score	Aptitude Score	Computer fundamentals score	Coding Score	Output
121910310021	9.8	9.79	63	45	63	Good to go
121910310032	9.2	8.7	42	39	40	Low Performance-requires training
121910310052	9.5	9.6	56	45	60	Needs Practice
121910310002	10	8.9	75	60	66	Good to go

Table 5.2 *Testcases along with I/P and O/P.*

VI. Conclusion & Future

Through the course of this project, we explored different machine learning models, experimented with different algorithms and techniques, and identified the ones that worked best for this problem, but several studies have shown that a student's performance is influenced by a complex interplay of various factors, and that no single factor can fully explain a student's performance.

Additionally, it is vital to note that different students will be affected differently by various factors, so it's important to consider each student individually.

In terms of future scope, this project can be extended to include more features that are relevant to predicting student placement performance. For example, demographic factors such as gender, ethnicity, and socioeconomic status could be included in the model to see if they have an impact on performance. Additionally, natural language processing techniques [15] could be used to analyse student essays or other written materials to gain additional insights into their abilities and potential for success.

Another area of future research could be to use this model to predict student performance in other areas beyond placements [16], such as academic success or job performance. This would require collecting data on different metrics that are relevant to these areas and using them to train and test the model.

References

[1] Smith, J. (2021). How Machine Learning is Transforming Education. EdTech Magazine.

[2] Koc, E. (2017). The Effect of Internship on Career Success: Evidence from the Banking Sector. Journal of Education and Practice, 8(7), 87-95.

[3] Zhao, Y., & Wang, X. (2020). The impact of academic performance and career development on student performance in internships: An empirical study. Education and Training, 62(7/8), 825-838.

[4] Wong, W., Cheung, F., Li, Y., & Li, C. (2018). Predicting student performance in internships: The roles of extracurricular activities, academic performance, communication skills, time management, and adaptability. Journal of Education for Business, 93(5), 215-221.

[5] Chua, Y. P., Lim, Y. T., Tan, H. P., & Kow, Y. M. (2018). Employers' perceptions of interns' employability skills and factors affecting employability: An exploratory study. Asia Pacific Education Review, 19(4), 557-567.

[6] Tsai, W. T., Li, Y. H., Li, J. Y., & Chang, Y. H. (2019). Predictors of student performance in software development internships. Journal of Educational Computing Research, 57(2), 344-358.

[7] Kulkarni, V., Kamath, S., & Anandh, N. (2019). Predicting student performance in placements using machine learning algorithms. Journal of Intelligent Systems, 28(4), 667-681.

[8] Rasheed, M. A., Ahmed, A. M., & Hassan, S. U. (2020). Exploratory data analysis for machine learning: A review. Journal of Big Data, 7(1), 1-37.

[9] Brownlee, J. (2020). One Hot Encoding for Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

[10] Brownlee, J. (2020). How to Calculate Correlation Between Variables in Python. Machine Learning Mastery.

[11] Seaborn developers. (n.d.). seaborn.pairplot. Seaborn Documentation. https://seaborn.pydata.org/generated/seaborn.pairplot.html

[13] Raschka, S. (2018). Python Machine Learning: scikit-learn, and TensorFlow 2. Packt Publishing.Link: https://www.packtpub.com/product/python-machine-learning-third-edition/9781789955750

[14] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. https://link.springer.com/article/10.1023/A:1010933404324

[15] Hirschberg, J., & Manning, C. (2015). Advances in natural language processing. Science, 349(6245), 261-266.

[16] Gupta, S., & Thakur, S. (2016). Emotional intelligence and academic performance of engineering students: An empirical study. Education and Information Technologies, 21(5), 1145-1157.