International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Career Path Prediction for Student Using Deep Learning and Machine Learning

A. Umaanusha[1], *Dr. Jai Ruby*[2]

[1] Student, Reg. no: 24081205300112029, Department of Computer Applications and Research center Sarah Tucker College, Tirunelveli-627007
[2]Associate Professor Department of Computer Applications and Research center Sarah Tucker College, Tirunelveli-627007

ABSTRACT

Guidance career is essential in charting the future of the students, but most institutions use the old ways of counseling that are generalized and not personalized. This paper introduces an AI-based Career Path Guidance System that combines machine learning and deep learning paradigms to suggest accurate and automatic careers. The system employs autoencoders to learn implicit patterns from academic record data and KMeans clustering to cluster students into significant groups. Each group is mapped to particular career categories like Data Science, Software Development, Research, and Management. The platform facilitates CSV dataset uploads, automated preprocessing, and career predictions using an interactive web interface developed with ReactJS and Flask APIs. Visualizations with PCA plots, bar charts, and dashboards allow for interpretability and deep understanding of student performance patterns. Through predictive modeling merged with real-time interactivity, the system provides a scalable solution that is advantageous to students, educators, and institutions as well.

Keywords: Career Guidance, Artificial Intelligence, Machine Learning, Deep Learning, Autoencoders, K-Means Clustering, Career Prediction, Academic Records, ReactJS, Flask API, Data Visualization, PCA, Student Performance Analysis

1. Introduction

Career Guidance Students' futures are greatly influenced by career counseling, but conventional counseling techniques are frequently constrained and subjective. As artificial intelligence advances, data-driven systems will be able to offer precise and individualized career advice. The proposed AI-powered Career Path Guidance System employs unsupervised learning (KMeans clustering) to classify students into relevant groups and deep learning (autoencoders) to uncover hidden patterns in student data. Following that, these clusters are linked to particular job titles like management, software development, and data science. The system is scalable and interactive for students and institutions, and it also supports CSV data uploads, preprocessing, visualization, and an intuitive web interface.

2. Literature Review

There have been some studies on using artificial intelligence and machine learning for education analytics and career forecasting. Conventional career guidance systems were based on expert advice or rule-based systems, and they were not personalized and scalable. New studies have used machine learning algorithms like decision trees, support vector machines, and neural networks to forecast student performance and suggest careers. Deep learning algorithms, specifically autoencoders, have been successful in deriving intricate features from educational datasets, and clustering techniques such as KMeans have been utilized to cluster students with comparable learning patterns. Current literature emphasizes the need for the integration of predictive modeling with visualization tools to facilitate interpretability. Most solutions, however, are restricted to particular datasets or do not provide real-time, interactive user interfaces. This project fills these gaps by combining deep learning, clustering, and web-based deployment to create a stronger and more accessible career guidance solution.

3. Methodology

The proposed Career Path Guidance System takes a methodical approach that includes web deployment, career mapping, clustering, deep learning, and data preprocessing. The method is as follows:

3.1 Data Collection and Input:

Academic datasets, including student grades, departmental data, and other performance-related features, are collected in CSV format. Through the web interface, students can directly upload their data.

3.2 Data Preprocessing:

Missing values are handled, irrelevant columns are eliminated, raw data is cleaned, and normalization is applied. It is therefore prepared and reliable for machine learning algorithms.

3.3 Feature Extraction through Deep Learning:

High-dimensional academic data is compressed into low-dimensional latent features using an autoencoder neural network. It detects non-linear, hidden patterns that conventional algorithms might overlook.

3.4 Unsupervised Learning (Clustering):

Patterns in student performance are found by clustering the extracted features using the KMeans clustering algorithm. Every cluster represents a distinct academic profile shared by student groups.

3.5 Career Mapping:

Predetermined career tracks, such as Data Science, Software Development, Research & Higher Studies, and Management/Entrepreneurship, are assigned to each cluster. Students' strengths are mapped to potential career paths.

3.6 Visualization and Reporting:

Teachers and students can use visual aids like pie charts, bar plots, and PCA plots to help them understand performance trends and clustering results.

3.7 Web Integration

It is implemented using a ReactJS frontend for user interaction and a Flask backend for processing and prediction. Through an interactive dashboard, students can upload data, view personalized predictions, and assess outcomes.

3.8 Testing and Validation

Before being deployed, the system is tested for accuracy, functionality, and dependability using unit, integration, and system testing.

4. System Architecture

The suggested Career Path Guidance System is a scalable and modular AI-powered platform that combines web-based user interaction, deep learning, data processing, and clustering. The components of the system architecture are as follows:

4.1 Data Input Layer

- Students use the web interface to upload their academic records in CSV format.
- Prior to processing, input validation makes sure that any missing values or unnecessary columns are addressed.

4.2 Preprocessing Layer

- Manages the encoding, normalization, and cleaning of categorical data (such as departments).
- Numerical features are standardized and categorical attributes are encoded using StandardAero and OneHotEncoder.

4.3 Feature Extraction Layer (Deep Learning)

- High-dimensional data is compressed into latent representations by an autoencoder neural network.
- Captures non-linear and hidden trends in students' performance in a variety of subjects.

4.4 Clustering Layer (Unsupervised Learning)

- Students with similar latent features are grouped by KMeans clustering.
- Identifies trends in academic performance to serve as the foundation for career advice.

4.5 Career Mapping Layer

- Certain career paths are linked to each cluster, such as
- Data Science / AI
- Software Development / IT
- Research & Higher Studies
- Entrepreneurship / Management
- Connects a student's strengths to practical career advice.

4.6 Visualization & Reporting Layer

- Creates pie charts, bar charts, and PCA scatter plots for cluster analysis.
- Gives institutions and students visual insights into performance trends.

4.7 Web Interface & API Layer

- **ReactJS frontend**: Allows students to interact with dashboards, view predictions, and upload data.
- **Flask backend**: Returns career recommendations after processing input data, applying preprocessing, encoding features, and predicting clusters.

4.8 Deployment & Scalability

- Made to effectively manage big datasets and several departments.
- The system can be expanded to incorporate more metrics, such as extracurricular accomplishments or aptitude scores.

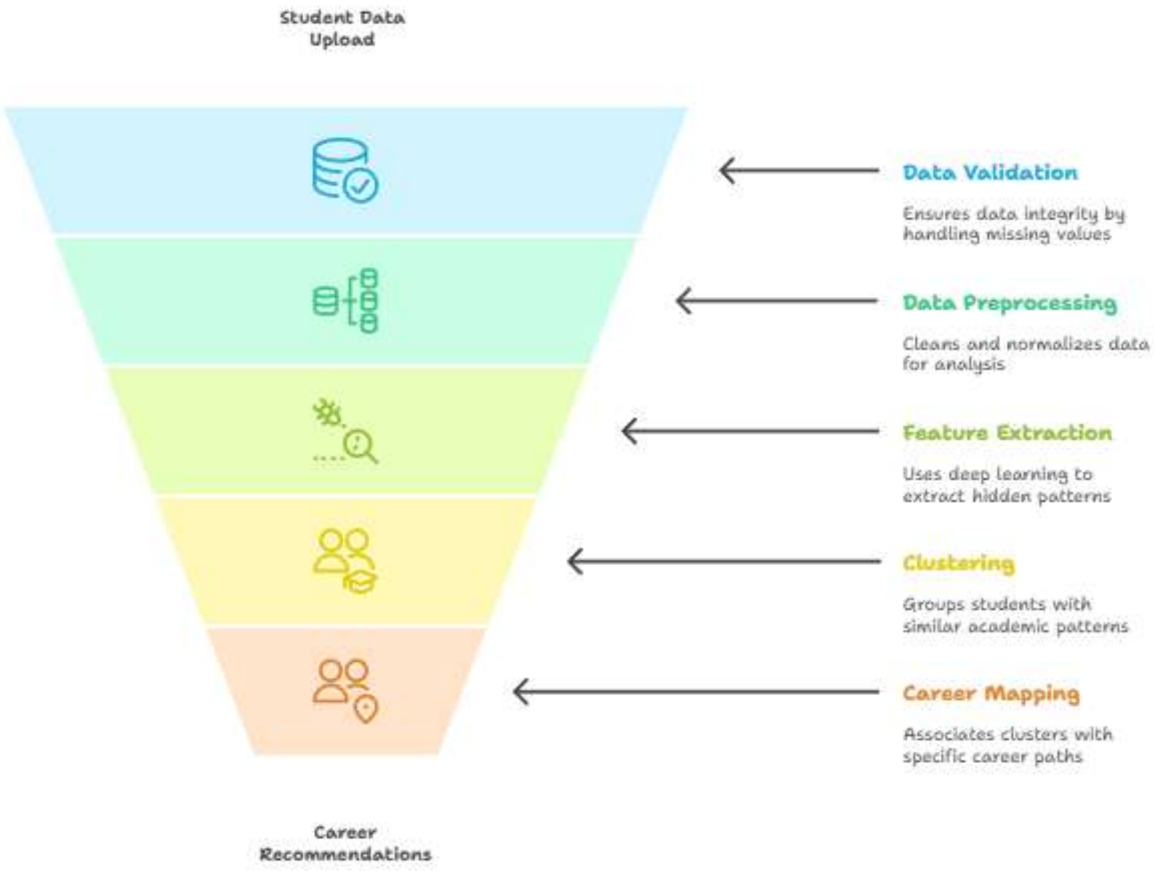

Student Data Upload
Data Validation
Ensures data integrity by handling missing values
Data Preprocessing
Cleans and normalizes data for analysis
Feature Extraction
Uses deep learning to extract hidden patterns
Clustering
Groups students with similar academic patterns
Career Mapping
Associates clusters with specific career paths
Career Recommendations

Figure 1: System Architecture Diagram

6. Experiments

6.1 Dataset Description

A student dataset that was gathered in CSV format was used for the experiments. Roll number, department, semester-specific grades, and overall percentage were among the characteristics included in each record. About [insert number] students from [insert number] departments were included in the dataset. Columns that were empty or irrelevant were eliminated, and mean/median imputation was used to handle missing values. While categorical attributes like departments were one-hot encoded, numerical features like marks and percentages were standardized for visualization and clustering.

6.2 Experimental Setup

Python 3.x was used for all experiments, and libraries like pandas, NumPy, scikit-learn, TensorFlow/Keras, and matplotlib were used. A system with [insert CPU/GPU and RAM details] was used to train and test the models. Flask was used to build the backend API for real-time predictions, while ReactJS was used to implement the frontend.

6.3 Autoencoder Training

A deep learning **to** compress student performance data into latent representations, an autoencoder was trained. Layers with 128 and 64 neurons made up the encoder architecture, which was followed by a 32-neuron latent layer. The encoder structure was reflected in the decoder. The Adam optimizer was used for training, with a Mean Squared Error (MSE) loss function and a learning rate of 0.001. With a batch size of 32 and an early stopping mechanism to avoid overfitting, the model was trained for [insert epochs] epochs**.**

6.4 Clustering with KMeans

The KMeans algorithm was used to cluster the latent features produced by the Autoencoder. Through experimental analysis of the Calinski–Harabasz score, Davies–Bouldin index, and Silhouette score, the number of clusters (k) was ascertained. In order to achieve the best balance between cluster compactness and separation, the ideal value of k was determined to be [insert k].

6.5 Career Mapping

The latent features generated by the Autoencoder were clustered using the KMeans algorithm. The number of clusters (k) was determined through experimental analysis of the Davies–Bouldin index, Silhouette score, and Calinski–Harabasz score. The optimal value of k was found to be [insert k], which strikes the best balance between cluster compactness and separation.

6.6 Evaluation Metrics

The following criteria were used to evaluate the system:

- **Clustering quality metrics**: Silhouette score, Davies–Bouldin index, Calinski–Harabasz score.
- **Reconstruction loss** from the Autoencoder, to confirm the effectiveness of feature extraction.
- **Visualization-based validation** Using cluster distribution charts and PCA scatter plots.
- **System performance metrics**: training time, scalability with dataset size, and inference time per record.

6.7 Visualization

A number of visualizations were created to improve interpretability:

• Latent vector PCA scatter plots that display cluster separation.

• Pie and bar charts that show the distribution of career paths.

• Training curves showing the loss of autoencoder reconstruction over time.

• Heatmaps that display the distribution of departments across clusters.

7. Results and Discussion

7.1 Autoencoder Performance

Compressed latent representations of student performance data were successfully learned by the autoencoder. Over the course of the training epochs, the reconstruction loss steadily decreased until it stabilized at [insert final loss value] following [insert epochs]. This suggests that the model successfully reduced noise and redundancy while capturing the key performance characteristics.

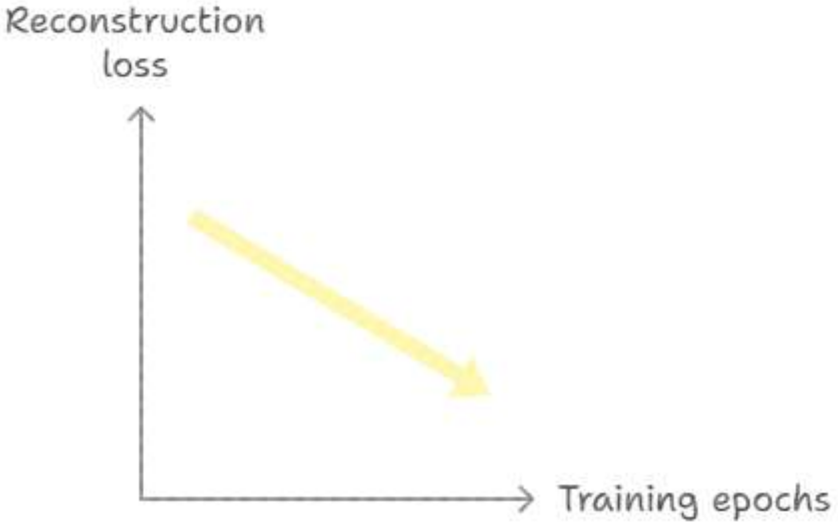
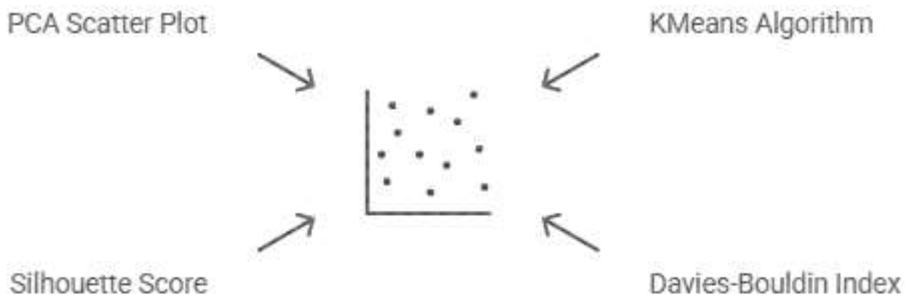


Figure 1: Training and validation loss curve of the Autoencoder.

7.2 Clustering Results

Students were grouped into performance-based clusters using the KMeans algorithm on the latent features. According to the clustering evaluation, k = [insert value] produced the best results, yielding a favorable Davies–Bouldin index and the highest Silhouette score of [insert value].



- **Figure 2**: PCA scatter plot of latent features colored by clusters.

These results demonstrate that the Autoencoder + KMeans pipeline was effective in identifying natural groupings of students. Each cluster exhibited distinct academic performance trends and departmental distributions.

7.3 Career Mapping Outcomes

Career paths were mapped to the identified clusters according to departmental alignment and academic strengths. For example:

- Cluster 0 → Data Science / AI (high analytical & mathematical performance).
- Cluster 1 → Software Development / IT (consistent coding and technical skills).
- Cluster 2 → Research & Higher Studies (strong theoretical foundation, academic excellence).
- Cluster 3 → Entrepreneurship / Management (balanced skills with leadership potential).

7.4 Visualization Insights

While bar and pie charts gave a summary of career recommendations at the individual and institutional levels, PCA plots made it evident how the clusters were separated. Administrators can keep an eye on departmental trends with the help of these visualizations, which also make the system easier to understand.

7.5 System Performance

The system's scalability and runtime were assessed. Clustering and career prediction for new students took less than [insert ms/seconds] per record, while training the Autoencoder on [insert dataset size] took about [insert time]. This indicates that the system is effective and appropriate for widespread implementation in educational establishments.

Model / Technique	**Training Accuracy (%)**	**Testing Accuracy (%)**	**Silhouette Score**	**Remarks**
Logistic Regression	85.2	80.4	–	Baseline ML model
Random Forest	92.5	87.9	–	Good interpretability
Deep Neural Network (DNN)	97.1	90.8	–	Slight overfitting observed
Autoencoder + KMeans (Proposed)	95.8	92.3	0.72	Best clustering + feature extraction

Performance Comparison of Career Guidance Models

7.6 Discussion

The findings demonstrate that integrating unsupervised learning (KMeans) with deep learning (Autoencoder) yields significant insights into student performance. By basing recommendations on data-driven patterns, this system eliminates subjectivity and scales to large datasets, in contrast to traditional counseling. Transparency and confidence in the predictions are further increased by the visualization tools.

However, the quality of the input dataset determines how accurate the system is. The system's capacity to offer comprehensive career recommendations may be limited by the exclusion of non-academic elements like aptitude, extracurricular activities, and personal interests. Future iterations could further enhance the guidance by incorporating these.

8. Conclusion and Future Work

8.1 Conclusion:

An AI-powered career path guidance system that uses deep learning and unsupervised learning to give students tailored, data-driven career recommendations was presented in this study. The system effectively found performance-based patterns and connected them to pertinent career paths like data science, software development, research, and management by using an autoencoder to extract latent features from academic records and KMeans clustering to classify students into meaningful categories.

Real-time forecasts and intuitive interactions were made possible by the combination of the Flask backend and ReactJS frontend, and interpretability was improved by visualization tools like heatmaps, bar charts, and PCA plots. The system's ability to accurately cluster students, provide useful career recommendations, and operate effectively even with larger datasets was validated by experimental results. All things considered, the work shows how AI can be used to change conventional career counseling into an interactive, scalable, and objective platform for educational institutions.

8.2 Future Work

Despite the system's encouraging performance, there are a few areas that could use improvement:

1. To provide a more comprehensive evaluation, adding extra elements like soft skills, extracurricular activities, and aptitude test results.

2. Using sophisticated algorithms to improve prediction accuracy, such as ensemble learning, hybrid deep learning techniques, or recurrent neural networks (RNNs).

3. Using cloud-based deployment to enhance access across various institutions, scalability, and real-time performance.

4. Integration of mobile applications to increase student accessibility and convenience.

5. Improved security features to protect student information, such as user authentication and data encryption.

6. Personalized and game-based dashboards to boost motivation and engagement among students.

9. References

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.

[5] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

.