Problem Statement:

Predicting student grades accurately is crucial for educators and institutions to provide personalized learning experiences and interventions. However, traditional methods often lack accuracy and efficiency. Therefore, there's a need for a machine learning-based system to predict student grades effectively.

Proposed System and Solutions:

The proposed system aims to leverage machine learning algorithms to predict student grades based on various factors such as attendance, past performance, study habits, etc. Solutions include data preprocessing to handle missing values and outliers, feature selection to identify relevant predictors, model training using algorithms like Random Forest, Support Vector Machines, or Gradient Boosting, and model evaluation to ensure accuracy and generalization.

System Development Approach:

The system development will follow an iterative approach involving stages like data collection, data preprocessing, feature engineering, model selection and training, model evaluation, and deployment. Agile methodologies could be employed for efficient development and continuous improvement.

Algorithm and Deployment:

For student grade prediction, algorithms like Random Forest, Support Vector Machines, or Gradient Boosting can be utilized due to their effectiveness in handling complex datasets and providing accurate predictions. After model training and evaluation, the system can be deployed either as a web application or integrated into existing educational platforms for easy access and utilization by educators.

Source code :

import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.utils import shuffle

data = pd.read\_csv("student-mat.csv")

data.head()

data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]

predict = "G3"

x = np.array(data.drop([predict], 1))

y = np.array(data[predict])

from sklearn.model\_selection import train\_test\_split

xtrain, xtest, ytrain, ytest = train\_test\_split(x, y, test\_size=0.2)

linear\_regression = LinearRegression()

linear\_regression.fit(xtrain, ytrain)

accuracy = linear\_regression.score(xtest, ytest)

print(accuracy)

predictions = linear\_regression.predict(xtest)

for i in range(len(predictions)):

print(predictions[x], xtest[x], [ytest[x]])

Output :

[[16.16395534 14.23423176 14.08532841 5.28096434 14.23423176]

[16.16395534 16.16395534 14.08532841 5.28096434 7.97291422]

[14.52779998 11.92149651 14.08532841 9.13993948 4.71694746]

...

[ 4.71694746 11.92149651 3.9451298 9.13993948 9.13993948]

[12.56424351 4.92497623 3.9451298 5.28096434 5.28096434]

[11.92149651 9.05247158 3.9451298 5.28096434 16.16395534]] [[[15 16 2 0 2]

[15 14 2 0 2]

[15 14 3 0 6]

[ 7 6 2 0 10]

[15 14 2 0 2]]....

Result:

The performance of the deployed system can be evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, techniques like cross-validation and grid search can be employed to fine-tune the model for better performance. The results should demonstrate significant improvement over traditional methods in predicting student grades.

Conclusion:

In conclusion, the proposed machine learning-based system offers a reliable and efficient approach to predict student grades accurately. By leveraging advanced algorithms and techniques, educators and institutions can better understand student performance and tailor interventions accordingly, ultimately improving educational outcomes.

References:

[1] R. S. Michalski, "Machine Learning: An Artificial Intelligence Approach," Springer Science & Business Media, 2013.

[2] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer Science & Business Media, 2009.

[3] J. W. Lloyd, "Machine Learning for Predicting Student Outcomes," EDUCAUSE Review, vol. 51, no. 4, 2016.