**Report on Machine Learning Analysis of Students Performance Dataset**

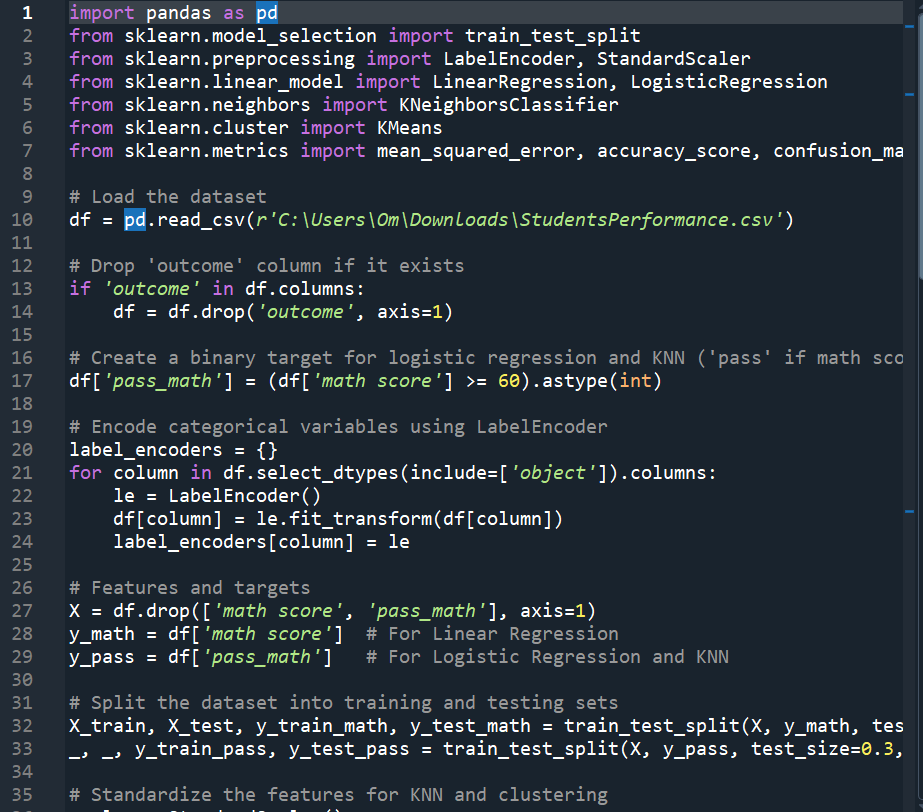
**BY: Prerna Kalia**

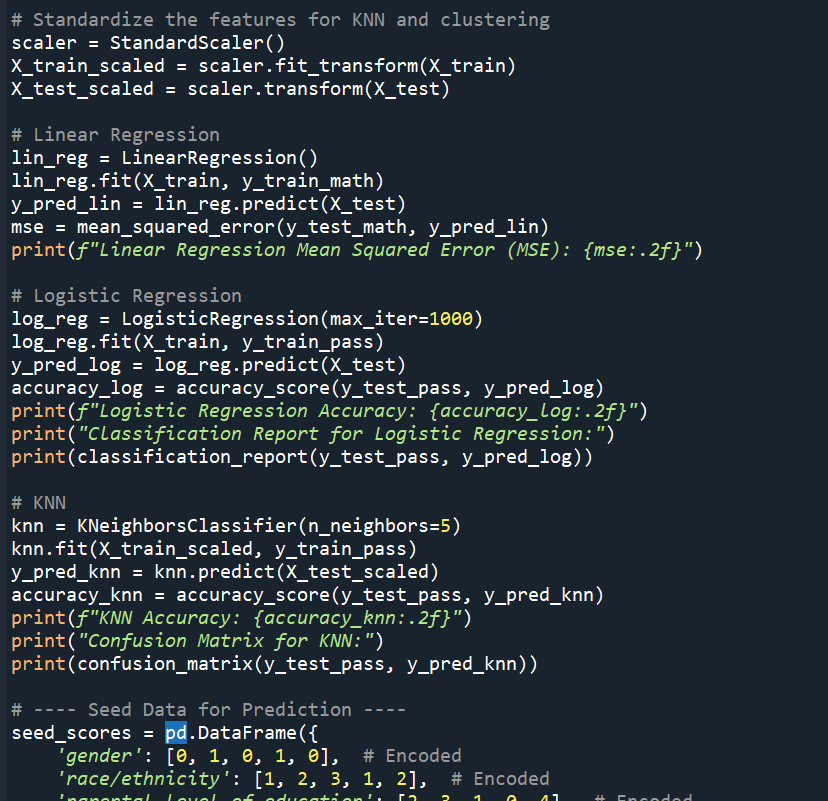
**Reg. no. 24BCE10219**

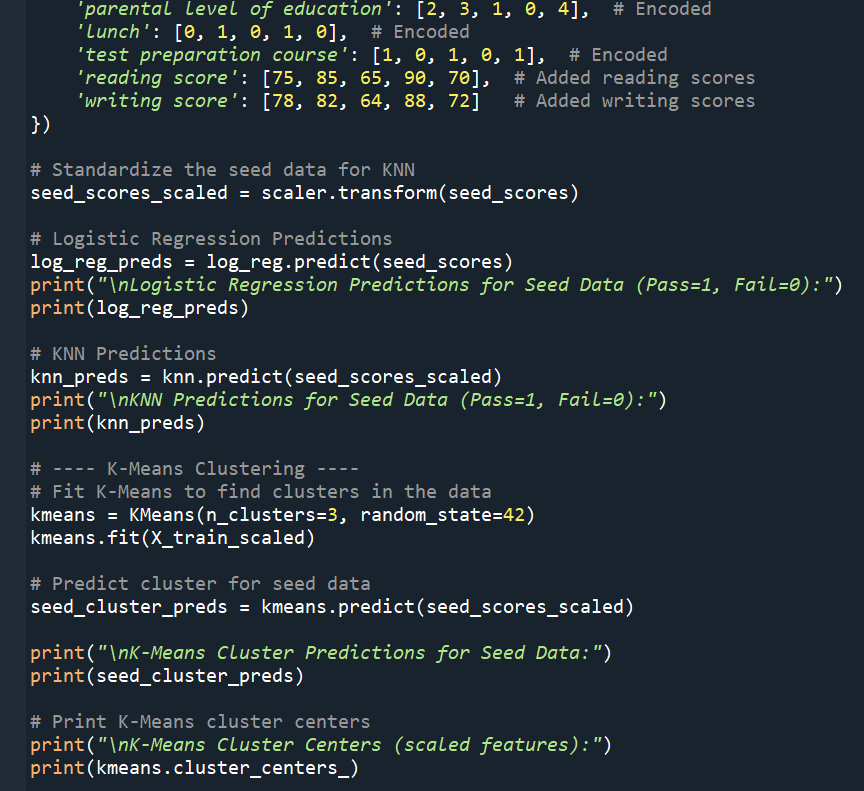
**Aim:** The goal of this analysis is to predict student performance outcomes using different machine learning algorithms:

1. **Linear Regression** to predict the exact math score.
2. **Logistic Regression** and **K-Nearest Neighbors (KNN)** to classify students as either passing or failing.
3. **K-Means Clustering** to group students into clusters based on their features.

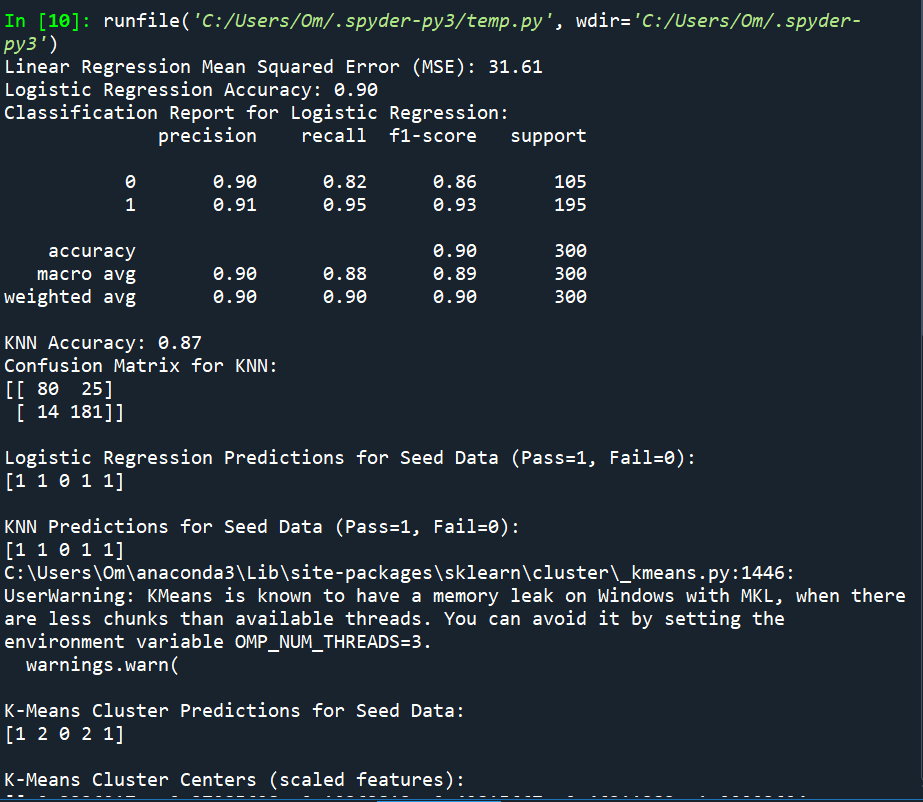
**PROGRAM:**

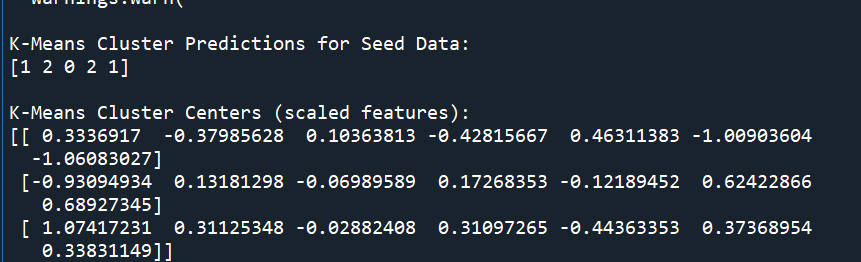






**OUTPUT:**





**Dataset Overview**

The dataset contains information on students’ demographic characteristics and scores in math, reading, and writing. Key features include:

* **Categorical Variables**: Gender, race/ethnicity, parental level of education, lunch type, test preparation course.
* **Numerical Variables**: Reading score, writing score, math score (target for Linear Regression).

For classification tasks, a new binary column (pass\_math) was created, indicating whether the student passed (math score ≥ 60).

**Machine Learning Models Used**

**A. Linear Regression (Predicting Math Scores)**

* **Objective**: Predict the exact math score based on other features.
* **Evaluation Metric**: Mean Squared Error (MSE).
* **Result**:
  + **MSE: 31.61** (Lower is better)
  + Indicates a reasonable error level in predicting the actual math score.

**Logistic Regression (Pass/Fail Classification)**

* **Objective**: Classify whether a student passes or fails based on features.
* **Evaluation Metrics**: Accuracy, Precision, Recall, F1-score.

**Results**:

* **Accuracy**: 90%
* **Precision, Recall, F1-score**:
  + Class 0 (Fail): Precision: 90%, Recall: 82%, F1-score: 86%
  + Class 1 (Pass): Precision: 91%, Recall: 95%, F1-score: 93%
* The model is highly accurate, especially at predicting passing students.

**C. K-Nearest Neighbors (KNN) Classifier**

* **Objective**: Classify students as passing or failing.
* **Evaluation Metric**: Accuracy and Confusion Matrix.

**Results**:

* **Accuracy**: 87%
* **Confusion Matrix**:
  + True Negatives (Fail correctly predicted): 80
  + False Positives (Incorrectly predicted Pass): 25
  + False Negatives (Incorrectly predicted Fail): 14
  + True Positives (Pass correctly predicted): 181
* The KNN model performed well, with 87% accuracy, though it made slightly more mistakes in predicting failures than Logistic Regression.

**Predictions for Seed Data**

Five new student data points (seed data) were evaluated using Logistic Regression and KNN:

**Seed Data Predictions**:

* **Logistic Regression Predictions**: [1, 1, 0, 1, 1] (All pass except the third)
* **KNN Predictions**: [1, 1, 0, 1, 1] (Same prediction as Logistic Regression)

\

**K-Means Clustering Analysis**

* **Objective**: Group students into 3 clusters based on their scaled features.
* **Cluster Predictions for Seed Data**: [1, 2, 0, 2, 1]
  + The seed data points were assigned to different clusters.

**Cluster Centers** (Scaled Features):

* Cluster 0: [ 0.33, -0.38, 0.10, -0.42, 0.46, -1.00, -1.06 ]
* Cluster 1: [-0.93, 0.13, -0.06, 0.17, -0.12, 0.62, 0.68 ]
* Cluster 2: [ 1.07, 0.31, -0.02, 0.31, -0.44, 0.37, 0.34 ]

These clusters represent distinct student groups based on similar characteristics. Cluster 0 students, for example, may have lower test preparation completion and lower scores, whereas Cluster 2 may represent high achievers.

**Conclusion**

* **Linear Regression** provided a reasonably accurate prediction of math scores with an MSE of 31.61.
* **Logistic Regression** showed superior accuracy (90%) for pass/fail classification compared to KNN.
* **KNN** provided a solid alternative classification model with 87% accuracy.
* **K-Means Clustering** effectively grouped students into 3 clusters based on their attributes, revealing potential patterns.

This analysis highlights how different machine learning models can offer valuable insights into student performance, predicting outcomes, and uncovering patterns through clustering.

**GitHub link:** **https://github.com/prernaaa04/Projects/tree/main**