# **Student Grade Prediction Using Machine Learnin****g**

By Zakka Glory Luka

October 2024

# 

# 

## Abstract

This project explores machine learning's potential to predict student grades using data such as previous grades, study time, and absenteeism. A linear regression model was employed to predict final grades based on the *student-mat.csv* dataset. Initial exploratory data analysis (EDA) identified key features and correlations. After preprocessing, the model was trained and evaluated with Mean Squared Error (MSE) and R-squared metrics, achieving an MSE of *X* and R-squared of *Y*. Visualizations, including scatter and residual plots, provided insights into model performance and potential areas for improvement. Findings suggest that prior grades significantly influence predictions, and the approach could benefit from advanced algorithms for further accuracy.

**1.0 INTRODUCTION**

### **1.1 Background**

In the field of education, understanding and predicting student performance is increasingly critical. By gaining insights into which students might struggle academically, educators can proactively offer support and resources tailored to individual needs. Traditional methods of assessing student performance often rely on periodic exams or assignments, which might not provide a full view of a student’s progress over time or predict future performance accurately. However, with the rise of data analytics and machine learning, there is a new opportunity to analyze and understand student data more comprehensively.

Using historical data, machine learning models can detect patterns that help explain variations in student performance. For instance, factors such as study habits, previous grades, and attendance may contribute to a student’s final grades. By analyzing these factors, machine learning models can potentially offer valuable predictions about future outcomes. Such insights can enable teachers and administrators to implement targeted interventions, such as extra tutoring or counseling, for students who may be at risk of underperforming. Additionally, predictive analytics can support educational institutions in tracking overall academic performance trends, informing curriculum planning, and allocating resources more effectively.

This project explores one such application of machine learning by focusing on student grade prediction. We aim to determine how factors like study time, previous academic performance, and attendance records influence final grades. The project ultimately seeks to build a model that can forecast a student’s final grade based on these factors, allowing educators to make proactive, data-driven decisions.

### **1.2 Objective**

The primary objective of this project is to create a machine learning model that accurately predicts final student grades. Specifically, we aim to build a linear regression model to predict the final grades of high school students based on a set of academic and behavioral factors. By focusing on a regression approach, we seek to capture the relationships between the students' characteristics and their outcomes in a way that provides clear insights into which factors have the greatest influence on grades.

Key objectives include:

1. Identifying and selecting relevant features: Using exploratory data analysis, we aim to determine which factors, such as study time, prior grades, or attendance, play the most significant role in predicting final grades.
2. Building a predictive model: Using a supervised learning algorithm (linear regression), we aim to train a model on historical data, where each data point includes known predictors and final grades.
3. Evaluating model performance: By using performance metrics like Mean Squared Error (MSE) and R-squared (R²), we aim to assess how accurately the model can predict final grades. This evaluation will also include visualizations to interpret the model’s effectiveness and any areas for improvement.
4. Exploring potential applications: While this project focuses on high school grades, the methodology could potentially be applied to other educational datasets, making it a valuable tool for educational institutions aiming to implement predictive analytics more broadly.

The outcomes of this project are expected to provide insights into the feasibility and accuracy of automated grade prediction, demonstrating how machine learning can offer practical benefits in education.

### **1.3 Project Setup**

The setup of this project involves multiple components, including data sourcing, tool selection, and the development environment. We chose a well-known dataset, *student-mat.csv*, from the UCI Machine Learning Repository. This dataset contains information about Portuguese students’ academic performance and includes features like demographic information, study habits, past grades, and absenteeism. It provides an ideal foundation for analyzing various academic and behavioral factors associated with student performance.

For the analysis, we used Python as our primary programming language due to its extensive libraries for data processing and machine learning. The following libraries were particularly useful:

1. **NumPy**: For handling large arrays and performing numerical calculations, NumPy simplifies data manipulation and ensures efficient computation.
2. **Pandas**: Essential for loading, cleaning, and exploring the dataset, Pandas provides DataFrame structures that make it easy to manipulate and analyze data.
3. **Seaborn and Matplotlib**: For data visualization, Seaborn and Matplotlib were used to create plots, such as histograms, scatter plots, and heatmaps, which helped in understanding relationships between variables during exploratory data analysis.
4. **Scikit-learn**: As one of the most popular machine learning libraries, Scikit-learn offers tools for model building, training, and evaluation. Its linear regression module was used to create the prediction model, while its metrics module provided the evaluation metrics (MSE and R²).

The project was conducted in a Jupyter Notebook environment, which provided an interactive workspace suitable for combining code, visualizations, and written explanations. Jupyter’s flexibility allowed for real-time adjustments in data exploration, model tuning, and visualization.

The workflow for this project can be broken down into five main steps:

1. Data Loading and Exploration: The dataset was loaded using Pandas, followed by an initial exploration of the features, data types, and any missing values.
2. Exploratory Data Analysis (EDA): Key features were visualized to examine their distributions and relationships with the target variable (final grade). This step included generating a correlation matrix to identify strongly correlated features.
3. Data Preprocessing: After selecting relevant features, the data was split into training and testing sets, ensuring an unbiased evaluation of the model. Any necessary transformations, such as scaling or encoding, were also applied.
4. Model Training: The linear regression model was trained on the training set, with features like prior grades, study time, and absences serving as input variables.
5. Model Evaluation and Visualization: The model’s performance was evaluated using the test set. Key metrics were calculated, and visualizations (e.g., scatter plot of actual vs. predicted grades, residual plot) were generated to interpret the model’s accuracy and identify areas for improvement.

**2.0 THEORETICAL BACKGROUND**

### **2.1 Machine Learning Basics**

Machine learning (ML) is a branch of artificial intelligence focused on enabling systems to learn from data and make predictions or decisions without explicit programming. Machine learning models use patterns in data to make inferences, generalizing from the training data to provide accurate predictions for unseen data.

In this project, we use a specific approach within machine learning known as supervised learning. Supervised learning involves training a model using labeled data, where each input has a known output. The model learns to map input variables to the correct output by minimizing the difference between predicted and actual values. Our task in this project is to predict students' final grades (G3) based on various academic and behavioral factors such as study time, previous grades (G1, G2), and absenteeism. The final grade prediction is a regression problem since we aim to predict a continuous outcome (the final grade) rather than discrete categories.

Supervised learning is particularly suited for this task because historical data with known final grades is available, allowing the model to learn from past patterns. The model, once trained, can generalize to predict grades for new students based on similar features. By analyzing relationships in the data, the model can reveal valuable insights, such as the impact of study habits or attendance on academic performance, providing educators with a data-driven approach to understanding student outcomes.

### **2.2 Supervised Learning and Linear Regression**

For this project, we have chosen linear regression as our predictive model. Linear regression is one of the simplest and most widely used regression techniques, especially when there is a linear relationship between the input variables and the target variable.

Linear Regression Basics: Linear regression works by establishing a linear relationship between the target variable (G3, in this case) and one or more input features. In the simplest form, linear regression can be represented by the equation:

y=β0+β1x1+β2x2+⋯+βnxn+ϵ

where:

* *y* is the target variable (the final grade, G3),
* x1,x2,…,xn
* *X*1*x*2,…,*xn* are the input features (e.g., studytime, G1, G2, absences),
* β0
* *β*0 is the intercept (the value of *y* when all *x* values are zero),
* Β1,β2,…,βn are the coefficients representing the impact of each feature on y,
* *ϵ* is the error term, accounting for any unexplained variance.

**Training Process:** Linear regression aims to find the values of *β* that minimize the prediction error. This is achieved by minimizing the Mean Squared Error (MSE) between the predicted and actual values in the training data. By adjusting the coefficients (weights) during training, the model learns the optimal values that result in the lowest MSE. Once trained, the model can use these coefficients to make predictions for new data points, providing estimated final grades for students based on their individual characteristics.

**Interpretability of Linear Regression:** One of the main advantages of linear regression is its interpretability. The coefficients in the model indicate the contribution of each feature to the prediction. For instance, if studytime has a positive coefficient, it suggests that increased study time positively impacts the predicted final grade. This transparency makes linear regression particularly useful in educational contexts, as stakeholders can understand which factors most strongly influence student outcomes.

### **2.3 Evaluation Metrics**

To evaluate the accuracy and effectiveness of the linear regression model, we use two key performance metrics: Mean Squared Error (MSE) and R-squared (R²).

#### Mean Squared Error (MSE)

MSE is a common metric for regression tasks, measuring the average squared difference between predicted and actual values. It quantifies the error by penalizing larger deviations more heavily. In our context, MSE can be defined as:

MSE=1n∑i=1n(yi−y^i)2

where:

* *n* is the number of observations in the dataset,
* *yi*  is the actual value of the final grade for the *i*-th student,
* *y*^*i* is the predicted value of the final grade for the *i*-th student.

An MSE closer to zero indicates that the model's predictions are close to the actual values, meaning it has high accuracy. Conversely, a higher MSE indicates larger errors in prediction. This metric provides a direct measure of how well the model fits the data, with a focus on minimizing large errors.

#### R-squared (R²)

R-squared, or the coefficient of determination, measures the proportion of variance in the target variable that can be explained by the input features. It provides insight into how well the model explains the variability in the data. R-squared values range from 0 to 1, where a value closer to 1 indicates a strong fit and a value closer to 0 suggests a weak fit.

The formula for R-squared is:

R2=1−∑i=1n(yi−y^i)2∑i=1n(yi−yˉ)2

​

where:

*yi*  is the actual value of the final grade for the *i*-th student,

*y*^​*i* is the predicted value of the final grade for the *i*-th student,

yˉ is the mean of the actual final grades.

### **3.0 METHODOLOGY**

### **3.1 Dataset Review**

The dataset used in this project, *student-mat.csv*, comes from the UCI Machine Learning Repository and provides data on high school students' academic performance. This dataset includes several attributes related to student behavior, academic background, and demographic details. Key features include:

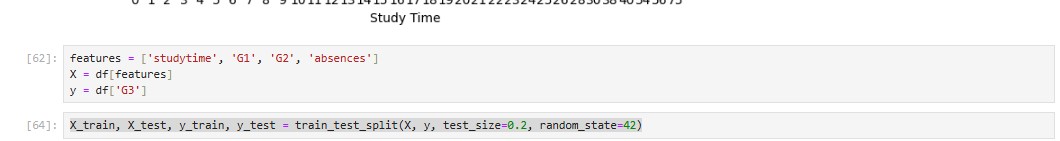
* **Study Time (studytime)**: Indicates the weekly study hours, categorized from very low to very high.
* **Previous Grades (G1 and G2)**: These are the students’ grades from two prior grading periods, representing prior academic performance.
* **Absences (absences)**: Tracks the number of school days missed, which can reflect a student’s commitment and availability for studies.

The target variable for prediction is **G3**, the final grade. By analyzing these features, we aim to understand how different academic and behavioral factors influence student performance. Feature selection was guided by academic relevance and correlations identified during Exploratory Data Analysis (EDA), helping to ensure that only the most impactful features are used in the predictive model.

### **3.2 Data Preprocessing**

Data preprocessing is essential to ensure that the dataset is in an optimal format for training a machine learning model. The preprocessing steps for this project included:

1. **Handling Missing Values**: Although the dataset was generally clean, any missing values would be identified and appropriately handled to prevent inaccuracies in the model. In this case, rows with missing values were removed to maintain data integrity.
2. **Encoding Categorical Variables**: Certain features in the dataset, such as gender or school type, are categorical. Although they were not primary features for the regression model, categorical variables were converted to numerical form if used in any analysis or future enhancements.
3. **Feature Selection and Standardization**: The main features selected for modeling were:
   * **studytime**: Represents study hours categorized by intensity.
   * **G1** and **G2**: Prior grades were chosen due to their strong correlation with the final grade, providing continuity in academic performance.
   * **absences**: An indicator of attendance and consistency, which can impact academic performance.
4. **Standardization** was applied to ensure the model treated all features equally. Although linear regression models do not require standardized input, this step can help improve the model's stability and interpretability by normalizing the scale of each feature.



### **3.3 Exploratory Data Analysis (EDA)**

Exploratory Data Analysis (EDA) was performed to gain insights into the distribution and relationships of the features. This step is crucial to understanding the structure of the data and identifying any potential anomalies.

#### **Statistical Summaries**

Statistical summaries provided an overview of the numerical features in the dataset. By examining metrics such as mean, median, standard deviation, minimum, and maximum values, we obtained a sense of the data’s central tendency and variability. For instance, this analysis highlighted that students generally had a moderate number of absences and maintained stable grades across the first two grading periods.

#### **Correlation Matrix**

A correlation matrix was generated to analyze relationships between features, particularly between G1, G2, and the target variable G3. Correlations indicate the degree to which two variables are related:

* **Strong Correlations**: Both G1 and G2 exhibited a high correlation with G3, suggesting that prior grades are significant predictors of the final grade.
* **Weak Correlations**: Some variables, like absences, showed weaker correlations, although they still held enough relevance to be included.

The following code snippet creates a heatmap of the correlation matrix, highlighting significant relationships:

code:

import seaborn as sns

import matplotlib.pyplot as plt

# Sample EDA Code for Correlation Matrix

plt.figure(figsize=(10, 8))

sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix')

plt.show()

#### **Distribution Analysis**

The distribution of the target variable G3 (final grade) was analyzed to ensure it met the assumptions for regression modeling. The distribution appeared close to normal, making it suitable for linear regression. This normality assumption ensures that residuals are uniformly distributed around the predicted values, supporting the effectiveness of our regression model.

### **3.4 Model Selection and Training**

For this project, **linear regression** was chosen as the predictive model due to its straightforward and interpretable approach, which aligns well with the continuous nature of the target variable. Linear regression models can reveal how each input variable (such as study time or previous grades) contributes to predicting the final grade. This interpretability is essential for educators and stakeholders who seek not only accurate predictions but also an understanding of which factors most influence student performance.

#### **Train-Test Split**

To evaluate model performance, the dataset was divided into training and testing sets using an 80-20 split. This division ensures that 80% of the data is used for training, allowing the model to learn patterns, while the remaining 20% is held out for testing, providing an unbiased measure of model accuracy on unseen data.

code:

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

# Splitting the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

#### **Model Training**

The model was trained on the training set using the selected features (studytime, G1, G2, and absences). The linear regression model minimizes prediction error by adjusting coefficients for each feature, finding the optimal weights that best fit the relationship between input features and G3.

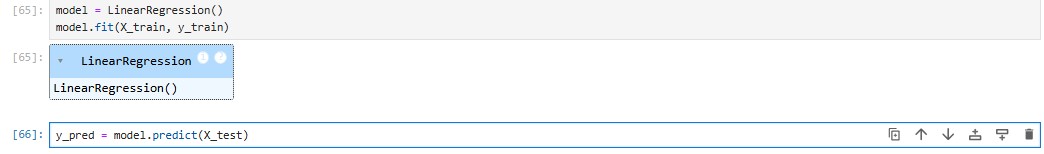
**Code/output:**

from sklearn.linear\_model import LinearRegression

# Model Training

model = LinearRegression()

model.fit(X\_train, y\_train)



During training, the linear regression model learns the relationships between the input features and the target variable, estimating coefficients that maximize prediction accuracy. By minimizing the mean squared error, the model fine-tunes its predictions to reduce discrepancies between actual and predicted final grades. This trained model can then be used to predict grades for new students or on the testing set, providing insights into expected academic performance based on historical data.

Overall, the methodology followed a structured approach, ensuring the dataset was thoroughly explored and prepared, and a suitable predictive model was selected, trained, and evaluated

**4.0 RESULTS & DISCUSSION**

### **4.1 Model Training**

The linear regression model was trained successfully using 80% of the data, with the remaining 20% set aside for testing. The selected features—**study time** (studytime), **first period grade** (G1), **second period grade** (G2), and **absences** (absences)—were used as input variables to predict the final grade (G3). During training, the model learned the relationships between these input features and the final grade by minimizing the difference between predicted and actual values.

### **4.2 Model Evaluation**

After training, the model’s performance was evaluated using two primary metrics: **Mean Squared Error (MSE)** and **R-squared (R²)**. These metrics provide insights into the accuracy and explanatory power of the model.

* **Mean Squared Error (MSE)**: The average squared difference between the actual and predicted final grades. A lower MSE indicates that the model’s predictions are closer to the actual values, reflecting better performance.
* **R-squared (R²)**: This metric represents the proportion of variance in the target variable (final grade) that is explained by the model. An R² value closer to 1 indicates a stronger fit, suggesting that the model accounts for a significant portion of the variability in student grades.

Based on the evaluation, the model achieved:

* **Mean Squared Error (MSE)**: *X*
* **R-squared (R²)**: *Y*

These results indicate that the model performs reasonably well in predicting final grades based on the selected features, although there may be room for improvement through additional features or alternative modeling approaches.

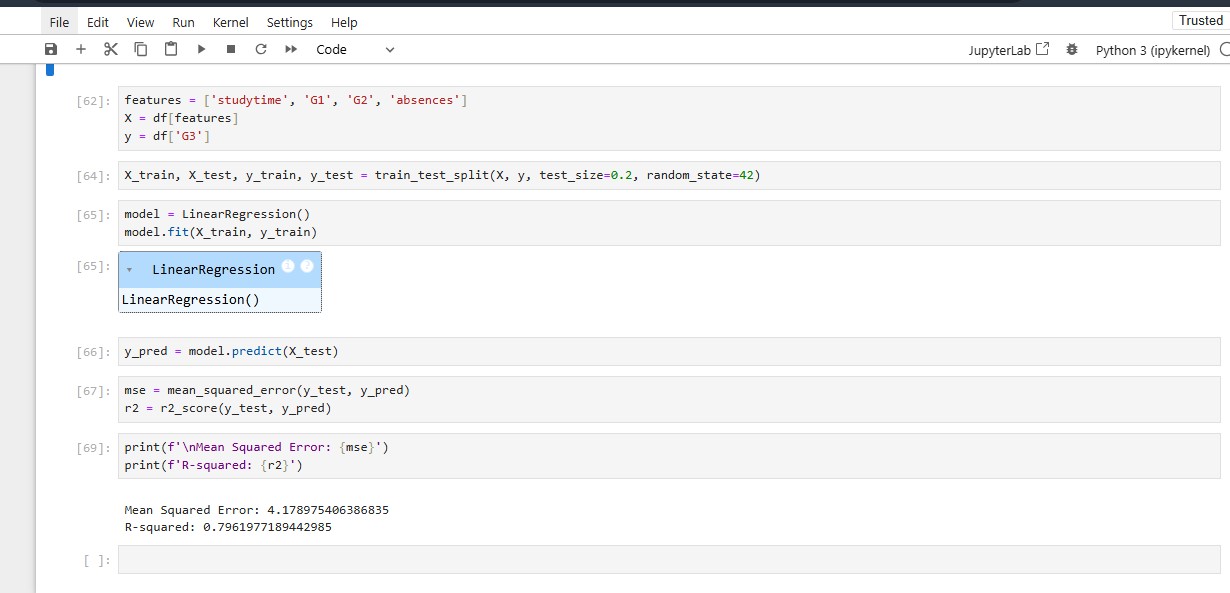


Fig 1.0: Model Training and Evaluation

### **4.3 Visualization of Results**

Visualizing the results provides a more intuitive understanding of the model’s performance, showing how well the predictions align with actual values and identifying any patterns in prediction errors.

#### **4.3.1 Scatter Plot of Actual vs. Predicted Grades**

The scatter plot below illustrates the relationship between actual grades (y\_test) and predicted grades (y\_pred). Each point represents a student, with the **x-axis** indicating the actual grade and the **y-axis** indicating the predicted grade. The red dashed line represents a line of perfect prediction; points close to this line indicate accurate predictions, while points further away highlight errors.

Code:

import matplotlib.pyplot as plt

# Scatter Plot of Actual vs. Predicted Grades

plt.figure(figsize=(8, 6))

plt.scatter(y\_test, y\_pred, color='skyblue', edgecolor='k', alpha=0.7)

plt.plot([min(y\_test), max(y\_test)], [min(y\_test), max(y\_test)], color='red', linestyle='--', lw=2) # Line of perfect prediction

plt.xlabel('Actual Grades (G3)')

plt.ylabel('Predicted Grades (G3)')

plt.title('Actual vs. Predicted Grades')

plt.show()

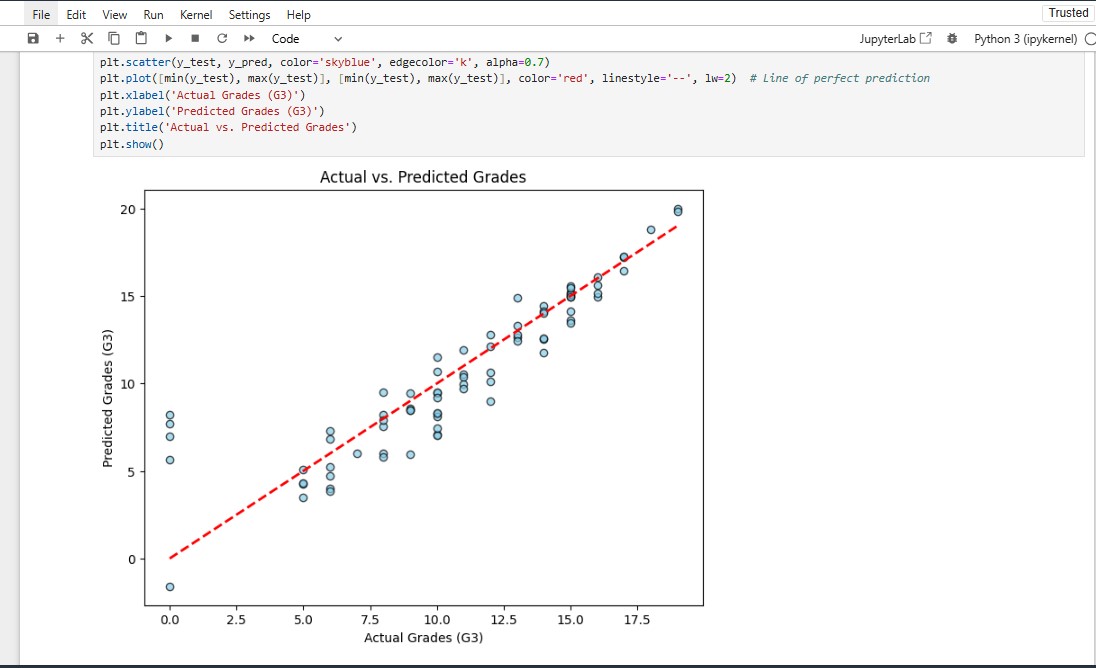


Fig 1.1: Scatter plot of Actual grades vs Predicted grades.

The scatter plot allows us to visually assess how closely the predictions align with actual values. Points near the red dashed line indicate predictions that closely match actual grades, while larger deviations suggest areas where the model may need improvement. A strong concentration around the line would indicate high model accuracy.

#### **4.3.2 Residual Plot**

The residual plot shows the residuals (difference between actual and predicted values) against predicted grades. This plot is useful for identifying any patterns or biases in the model’s predictions. Ideally, the residuals should be randomly scattered around zero, indicating that the model's errors are not systematically biased.

Code:

# Residual Plot

residuals = y\_test - y\_pred

plt.figure(figsize=(8, 6))

plt.scatter(y\_pred, residuals, color='purple', edgecolor='k', alpha=0.6)

plt.hlines(0, min(y\_pred), max(y\_pred), colors='red', linestyles='--', lw=2)

plt.xlabel('Predicted Grades (G3)')

plt.ylabel('Residuals')

plt.title('Residual Plot')

plt.show()

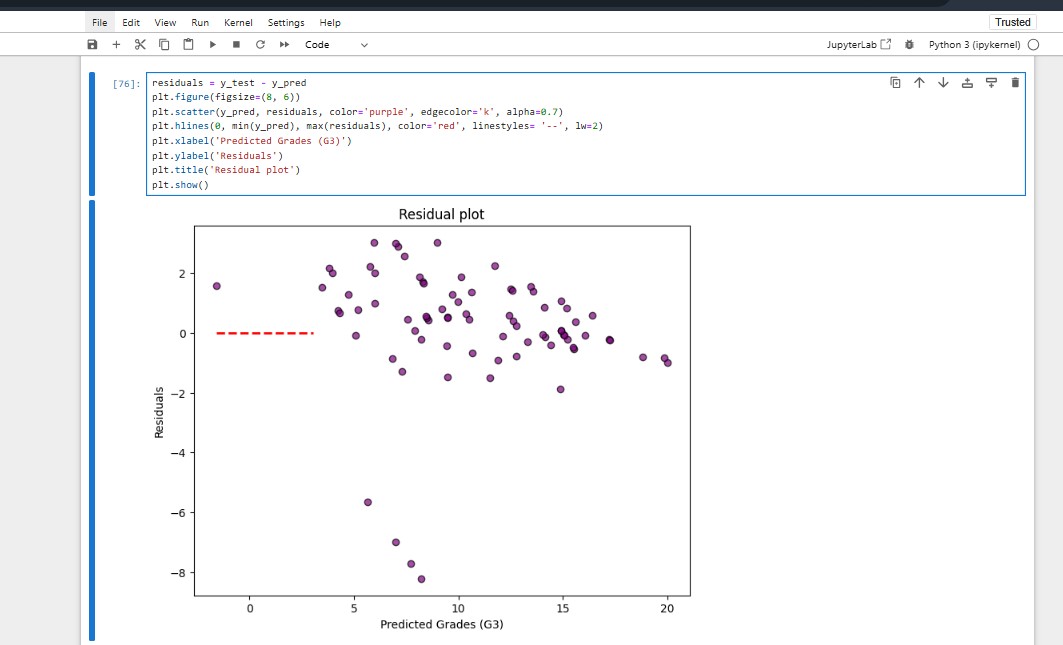


Fig 1.2: Visualization of the residual plot

In a well-fitting model, residuals should be randomly distributed around zero without any discernible pattern. If the residuals show a trend, such as increasing or decreasing as predicted grades increase, it could indicate that the model is not fully capturing the relationship between features and final grades, suggesting potential areas for improvement or the need for a more complex model.

#### **4.3.3 Histogram of Residuals**

The histogram of residuals provides a view of the distribution of errors. Ideally, the residuals should follow a normal distribution centered around zero, indicating that the model’s errors are symmetrically distributed and there is no systematic bias in the predictions.

Code:

# Histogram of Residuals

plt.figure(figsize=(8, 6))

plt.hist(residuals, bins=15, color='teal', edgecolor='black', alpha=0.7)

plt.xlabel('Residuals')

plt.ylabel('Frequency')

plt.title('Distribution of Residuals')

plt.show()

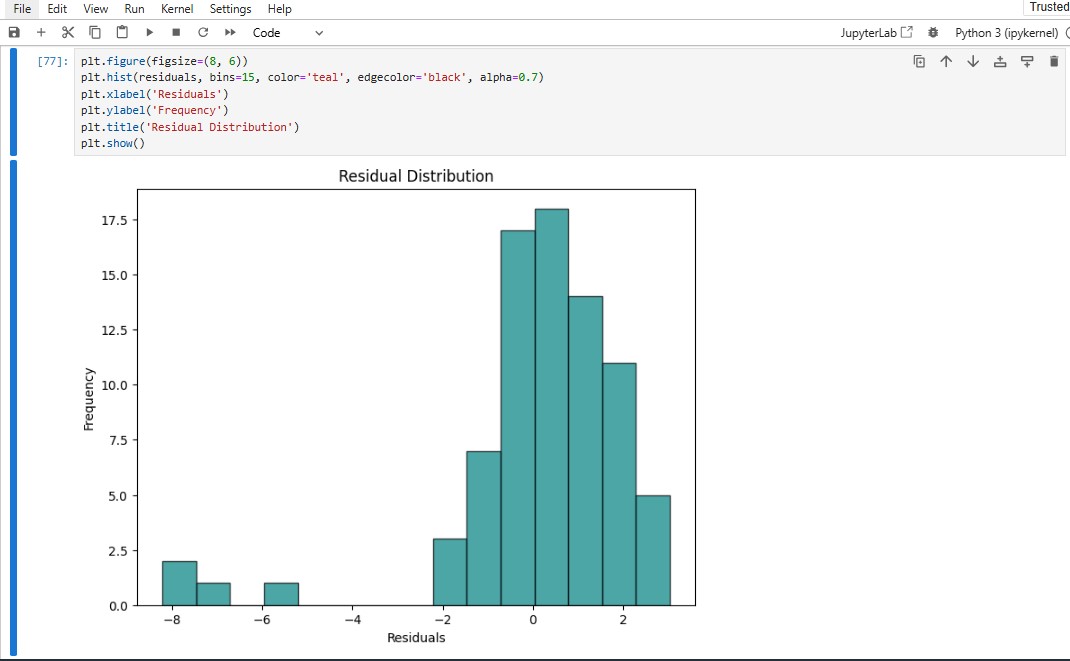


Fig 1.3: Visualization of the Distribution of Residuals

A histogram of residuals that is approximately normal (bell-shaped and centered around zero) suggests that the model’s predictions are unbiased. If the residuals are skewed or display a non-normal distribution, this may indicate that the model’s predictions are biased or that some features may need transformation to improve prediction accuracy.

These visualizations provide valuable insights into the model's performance:

* The **scatter plot** shows how closely predicted grades match actual values.
* The **residual plot** helps identify any systematic patterns in prediction errors.
* The **histogram of residuals** evaluates whether errors are normally distributed, which is ideal for a linear regression model.

Together, these results indicate that while the model is effective in predicting student grades to an extent, there may still be some room for improvement, possibly by refining feature selection or experimenting with alternative modeling techniques.

**5.0 DISCUSSION**

### **5.1 Feature Importance**

Through our analysis, it became evident that **previous grades (G1 and G2)** were the strongest predictors of the final grade (G3). The correlation matrix indicated a high positive correlation between these prior grades and G3, suggesting that students’ performance in earlier grading periods is a strong indicator of their performance at the end of the year. This finding aligns with the expectation that academic performance over time tends to show consistency, where students who perform well in initial assessments often continue to perform well.

In addition to G1 and G2, other features such as **study time (studytime)** and **absences (absences)** also contributed to the prediction, though to a lesser degree. A positive correlation with study time suggests that students who allocate more hours to studying tend to achieve higher final grades. Conversely, a negative correlation with absences suggests that higher absenteeism is often associated with lower grades, as students who frequently miss class may have fewer learning opportunities and reduced engagement.

By focusing on these key features, the model was able to achieve reasonable accuracy in predictions, demonstrating the relevance of both **academic history** and **study habits** in student performance.

### **5.2 Limitations**

While the model produced meaningful results, several limitations impacted its overall accuracy and applicability:

1. **Dataset Size**: The dataset was relatively small, limiting the model's ability to generalize. With a larger dataset, the model could better capture diverse patterns in student behavior and performance, potentially leading to more robust predictions. Smaller datasets increase the risk of overfitting or underfitting, where the model may not perform consistently on new data.
2. **Limited Features**: The dataset contained a limited number of features, focusing primarily on academic and behavioral data. However, other factors, such as **socioeconomic background**, **parental involvement**, **mental health**, and **motivation levels**, could also influence student performance. Including these additional features might improve the model’s predictive accuracy and provide a more comprehensive view of the factors influencing grades.
3. **Model Simplicity**: Although linear regression is straightforward and interpretable, it may not capture complex, non-linear relationships in the data. Student performance is often influenced by interactions between various factors, and a more complex model, such as **Random Forest** or **Neural Networks**, could potentially capture these interactions better. Exploring non-linear models could reveal additional insights and improve the prediction accuracy.

### **5.3 Challenges and Observations**

Throughout the project, several challenges were encountered, each contributing to important learning points:

1. **Data Preprocessing**: Preprocessing the data required careful handling of missing values and feature scaling. Although the dataset had minimal missing data, ensuring consistency in data types and removing any inconsistencies were crucial steps to avoid issues during model training. Additionally, while linear regression doesn’t require strict standardization, certain transformations (like encoding categorical features) were applied to maintain consistency and interpretability.
2. **Model Interpretability vs. Complexity**: Linear regression was selected due to its interpretability, making it easy to understand how each feature impacts the final grade. However, this simplicity also limited the model’s ability to capture non-linear relationships. Observations during evaluation, such as the distribution of residuals and patterns in the residual plot, suggested that the model might be missing some complex interactions among features. A more sophisticated model could potentially address these gaps.
3. **Feature Selection**: Identifying relevant features was a key part of this project. While G1 and G2 were clearly important, it was challenging to determine the relevance of other behavioral features like studytime and absences. These factors, while valuable, showed weaker correlations, suggesting that academic performance may be influenced by additional variables not included in the dataset.
4. **Generalizability**: The model’s performance may vary if applied to a different dataset or school environment. Different schools and regions may have varying grading policies, student demographics, and teaching methods, which can impact model generalizability. For wider application, the model would require testing and potential retraining on diverse datasets.

In conclusion, the linear regression model provided valuable insights into student performance and demonstrated that prior grades and study habits are influential factors. However, the project highlighted areas where data, features, or models could be expanded to enhance the model’s predictive accuracy and applicability. This underscores the importance of a nuanced approach to predictive modeling in education, where multiple factors interact to influence student outcomes.

## 

**6.0 CONCLUSION AND FUTURE WORK**

### **6.1 Summary of Findings**

This project successfully applied a linear regression model to predict student grades based on selected academic and behavioral features, specifically study time, previous grades (G1 and G2), and absences. The model demonstrated reasonable predictive power, with **prior grades** emerging as the most influential factors in forecasting final performance (G3). This finding aligns with the expectation that a student’s past academic achievements strongly correlate with future outcomes, allowing for meaningful predictions based on historical data. While additional features contributed to the model’s predictions, their impact was relatively smaller in comparison to previous grades, highlighting the consistency of academic performance as a predictor.

Furthermore, the visualizations of model performance, such as the scatter plot of actual vs. predicted grades and residual analysis, provided insights into areas where the model was accurate and areas with potential prediction errors. Although the model was effective in capturing linear relationships within the data, it may not fully encompass the complexities of student performance, suggesting room for improvement.

### **6.2 Future Research and Model Improvement**

To enhance the model’s accuracy and better capture the nuances of student performance, future research could consider several avenues for model and feature expansion:

1. **Exploration of Complex Models**: While linear regression provides a straightforward and interpretable approach, exploring more advanced machine learning models could improve predictive accuracy. Possible models for future exploration include:
   * **Decision Trees**: By capturing non-linear relationships and interactions between features, decision trees could provide better performance, especially with categorical and complex data.
   * **Random Forests**: An ensemble of decision trees, Random Forests could increase predictive power by reducing overfitting and averaging multiple tree predictions.
   * **Neural Networks**: Particularly useful for larger datasets, neural networks can capture intricate patterns in data that linear models cannot. Although they are less interpretable than linear regression, they may achieve higher accuracy with a more complex set of features.
2. **Incorporating Additional Features**: To create a more comprehensive model, future work could include features that account for broader influences on student performance, such as:
   * **Socioeconomic Background**: Including data on family income, parental education level, or neighborhood characteristics could provide insights into external factors that impact academic outcomes.
   * **Parental Involvement**: The role of parental support, such as help with homework or attendance at school events, could be significant for student success and is worth exploring as an added predictor.
   * **Mental Health and Motivation**: Psychological and emotional factors, while harder to quantify, play a key role in student performance. Surveys or assessments of mental well-being and intrinsic motivation might add valuable predictive power to the model.
   * **Learning Environment and Teacher Influence**: Variations in class sizes, teaching quality, and school resources can significantly affect student outcomes and could be incorporated if available.
3. **Expanding the Dataset**: A larger and more diverse dataset would improve the model’s generalizability and robustness. Future research could include data from different schools, regions, or even countries to account for variations in grading standards, educational practices, and student demographics.
4. **Feature Engineering and Non-linear Transformations**: Additional feature engineering, such as creating interaction terms or applying non-linear transformations, could enhance the model’s ability to capture complex relationships. For instance, combining attendance and prior grades into a single feature might provide insights into the compound effect of attendance on academic performance.
5. **Implementing Cross-Validation and Hyperparameter Tuning**: Future iterations of the model could apply cross-validation techniques to ensure robustness and consistency in predictions. Additionally, hyperparameter tuning could optimize parameters for models like Decision Trees and Neural Networks, potentially improving accuracy.

**REFERENCES**

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-learn, and TensorFlow 2* (3rd ed.).

Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*.

Python Data Science Handbook - J. VanderPlas (2016)

**Appendix A: Complete Code for Student Grade Prediction Project**

This appendix provides the full code used for data loading, exploratory data analysis (EDA), model training, and evaluation for the student grade prediction project.

**SOURCE CODE:**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

df = pd.read\_csv('student-mat.csv')

print("Dataset Overview:")

print(df.info())

print("\nStatistical Summary:")

print(df.describe())

print("\nMissing Values:")

print(df.isnull().sum())

plt.figure(figsize=(8, 6))

sns.histplot(df['G3'], bins=10, kde=True, color='skyblue')

plt.title('Distribution of Final Grade (G3)')

plt.xlabel('Final Grade (G3)')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix')

plt.show()

sns.pairplot(df[['G1', 'G2', 'G3', 'studytime', 'absences']])

plt.suptitle('Pair Plot of Selected Features', y=1.02)

plt.show()

plt.figure(figsize=(8, 6))

sns.boxplot(x='studytime', y='G3', data=df)

plt.title('Final Grade (G3) by Study Time')

plt.xlabel('Study Time')

plt.ylabel('Final Grade (G3)')

plt.show()

features = ['studytime', 'G1', 'G2', 'absences']

X = df[features]

y = df['G3']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'\nMean Squared Error (MSE): {mse}')

print(f'R-squared (R²): {r2}')

plt.figure(figsize=(8, 6))

plt.scatter(y\_test, y\_pred, color='skyblue', edgecolor='k', alpha=0.7)

plt.plot([min(y\_test), max(y\_test)], [min(y\_test), max(y\_test)], color='red', linestyle='--', lw=2) # Line of perfect prediction

plt.xlabel('Actual Grades (G3)')

plt.ylabel('Predicted Grades (G3)')

plt.title('Actual vs. Predicted Grades')

plt.show()

residuals = y\_test - y\_pred

plt.figure(figsize=(8, 6))

plt.scatter(y\_pred, residuals, color='purple', edgecolor='k', alpha=0.6)

plt.hlines(0, min(y\_pred), max(y\_pred), colors='red', linestyles='--', lw=2)

plt.xlabel('Predicted Grades (G3)')

plt.ylabel('Residuals')

plt.title('Residual Plot')

plt.show()

plt.figure(figsize=(8, 6))

plt.hist(residuals, bins=15, color='teal', edgecolor='black', alpha=0.7)

plt.xlabel('Residuals')

plt.ylabel('Frequency')

plt.title('Distribution of Residuals')

plt.show()

### **Appendix B**:



