BIRZEIT UNIVERSITY

Faculty of Engineering Technology

Electrical & Computer Engineering Department

ENCS5341, Machine Learning and Data Science

Project Report

Prepared by:

Name (1): Mohammed Owda **ID Number:** 1200089

Name (2): Mohammed Abu Shams **ID Number:** 1200549

Instructor: Dr. Yazan Abu Farha

Date: 26.1.2024

Section: 2

Introduction

In this project, we're trying to figure out how well we can predict if students will pass or fail based on a bunch of information about them. This is very important because it can help schools identify students who might need extra help early on.

To address this task, we employed three different machine learning models:

K-Nearest Neighbors (KNN): This is like a simple starting point. It looks at nearby students to guess how a new student might do. We are changing how many nearby students it looks at.

Random Forest Classifier: This is like having a bunch of smart decision-makers working together. They build lots of little decision trees and combine their answers for a better guess.

Logistic Regression: This is a straightforward way of looking at things. It's good when the connection between the information and passing or failing is pretty straight.

For a comprehensive evaluation of the models, we used a variety of metrics including accuracy, precision, recall, F1-score, and confusion matrices. Accuracy measures the overall proportion of correct predictions, while precision and recall provide insight into the model's performance with respect to each class label. The F1-score offers a balance between precision and recall, which is particularly useful in the context of imbalanced classes. Confusion matrices give a detailed breakdown of the model's performance, highlighting the instances of true positives, false positives, true negatives, and false negatives.

Dataset

We choose a dataset about classification From Kaggle: https://www.kaggle.com/datasets. We pick Students Performance Dataset called ‘student-mat.csv’ it contains data related to student performance in mathematics.
This project uses information about 395 students with 33 different features and one target variable, 'G3', which indicates the final grade and is the basis for our binary classification task—pass or fail. The features cover things like age, gender, where they live, family size, their parents' education, how much time they spend studying, any previous failures, how they spend their free time, going out, and their grades in the first, second, and final periods (G1, G2, G3).

A total of 395 instances provided a robust foundation for applying and evaluating machine learning models. Before the application of these models, the dataset underwent a rigorous exploratory data analysis (EDA). This process involved generating descriptive statistics to capture the central tendencies and variabilities of the features, as well as employing visual tools

like histograms and box plots to depict the distributions and identify outliers or patterns within the data.

Our exploratory data analysis included generating descriptive statistics and visualizations to understand the data distribution. Key to our EDA was the inclusion of a correlation matrix, presented as a color-coded heatmap, to reveal the strength and direction of relationships between features and the target variable. This matrix was particularly useful for identifying predictive features, such as prior grades (G1 and G2), and understanding their influence on the final grade.

```
Dataset Description:
Number of Rows: 395
Number of Columns: 33

First 5 Rows of the Dataset:
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob      Fjob  ... \
0     GP   F   18       U     GT3       A     4     4  at_home   teacher  ...
1     GP   F   17       U     GT3       T     1     1  at_home     other  ...
2     GP   F   15       U     LE3       T     1     1  at_home     other  ...
3     GP   F   15       U     GT3       T     4     2   health  services  ...
4     GP   F   16       U     GT3       T     3     3    other     other  ...

   famrel  freetime  goout  Dalc  Walc  health  absences  G1  G2  G3
0       4         3      4     1     1       3         6   5   6   0
1       5         3      3     1     1       3         4   5   5   0
2       4         3      2     2     3       3        10   7   8   1
3       3         2      2     1     1       5         2  15  14   1
4       4         3      2     1     2       5         4   6  10   1

[5 rows x 33 columns]

Descriptive Statistics:
              age        Medu        Fedu  traveltime   studytime    failures  \
count  395.000000  395.000000  395.000000  395.000000  395.000000  395.000000
mean    16.696203    2.749367    2.521519    1.448101    2.035443    0.334177
std      1.276043    1.094735    1.088201    0.697505    0.839240    0.743651
min     15.000000    0.000000    0.000000    1.000000    1.000000    0.000000
25%     16.000000    2.000000    2.000000    1.000000    1.000000    0.000000
50%     17.000000    3.000000    2.000000    1.000000    2.000000    0.000000
75%     18.000000    4.000000    3.000000    2.000000    2.000000    0.000000
max     22.000000    4.000000    4.000000    4.000000    4.000000    3.000000

            famrel    freetime       goout        Dalc        Walc      health  \
count  395.000000  395.000000  395.000000  395.000000  395.000000  395.000000
mean     3.944304    3.235443    3.108861    1.481013    2.291139    3.554430
std      0.896659    0.998862    1.113278    0.890741    1.287897    1.390303
min      1.000000    1.000000    1.000000    1.000000    1.000000    1.000000
25%      4.000000    3.000000    2.000000    1.000000    1.000000    3.000000
50%      4.000000    3.000000    3.000000    1.000000    2.000000    4.000000
75%      5.000000    4.000000    4.000000    2.000000    3.000000    5.000000
max      5.000000    5.000000    5.000000    5.000000    5.000000    5.000000

         absences          G1          G2          G3
count  395.000000  395.000000  395.000000  395.000000
mean     5.708861   10.908861   10.713924    0.670886
std      8.003096    3.319195    3.761505    0.470487
min      0.000000    3.000000    0.000000    0.000000
25%      0.000000    8.000000    9.000000    0.000000
50%      4.000000   11.000000   11.000000    1.000000
75%      8.000000   13.000000   13.000000    1.000000
max     75.000000   19.000000   19.000000    1.000000
```

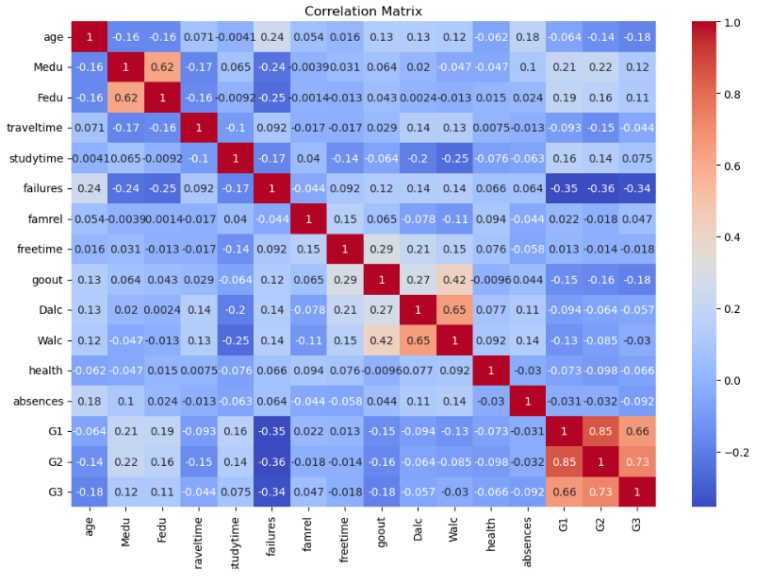

Figure 1

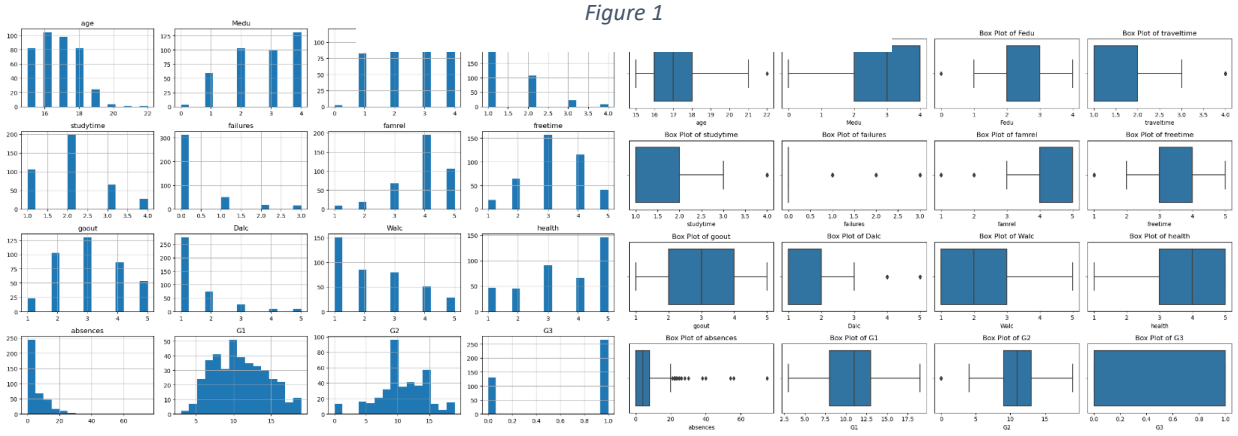

Figure 2

Experiment and Results

Prior to establishing our baseline model, we engaged in a crucial step of feature selection to streamline our dataset. From an initial set of 33 features, we utilized the Random Forest algorithm to identify and retain the top 10 features that most significantly contributed to predicting the final grade, 'G3'. This process was not only aimed at enhancing the models' performance by reducing dimensionality but also at minimizing the risk of overfitting and ensuring the models' focus on the most predictive attributes.

1. Baseline Model

We established a baseline model using the k-nearest neighbors (KNN) algorithm with two different values of k: 1 and 3. This approach provided a simple yet informative baseline against which we could measure the performance of more complex machine learning models. We chose Manhattan distance as our distance metric for its suitability in high-dimensional data spaces.
The accuracies were as follows:

```
=================== KNN 1 ===================
K1 Training Accuracy: 1.0
K1 Test Accuracy: 0.815

K1 Model Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.67      0.74        46
           1       0.81      0.90      0.86        73

    accuracy                           0.82       119
   macro avg       0.82      0.79      0.80       119
weighted avg       0.82      0.82      0.81       119

K1 Model Confusion Matrix:
[[31 15]
 [ 7 66]]
```

```
=================== KNN 3 ===================
K3 Training Accuracy: 0.935
K3 Test Accuracy: 0.832

K3 Model Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.67      0.76        46
           1       0.82      0.93      0.87        73

    accuracy                           0.83       119
   macro avg       0.84      0.80      0.81       119
weighted avg       0.84      0.83      0.83       119

K3 Model Confusion Matrix:
[[31 15]
 [ 5 68]]
```

Figure 3

For K=1:

The baseline model using the k-nearest neighbor algorithm with k=1 exhibited perfect training accuracy, indicating an overfit to the training data. The test accuracy was notably high at 81.5%, which is commendable for such a simple model, though the lower recall for class 0 (0.67) compared to class 1 (0.90) suggests a potential bias towards the majority class. But because it only looks at the one closest match, it might get thrown off by unusual or noisy data. The confusion matrix confirmed the model's tendency to favor class 1, with more false negatives for class 0.

For K=3:

When the number of neighbors was increased to k=3, the model's test accuracy slightly improved to 83.2%, and training accuracy remained high at 93.5%. This adjustment appeared to enhance the model's generalization capability, as evidenced by the improved recall for class 1 (0.93) and the consistent precision across both classes. The F1-scores and confusion matrix reflected a

balanced improvement in identifying class 1, with fewer false negatives compared to the k=1 model. This means it might be a good idea to check if looking at even more neighbors could make the model even better.

2. The proposed ML models

In pursuit of improved performance over the baseline nearest neighbor model, we evaluated two additional machine learning models: **Random Forest** and **Logistic Regression**. These models were selected based on their distinct characteristics and suitability for classification tasks.

Random Forest Classifier

Motivation for Selection:

Random Forest was chosen for its ability to handle complex datasets with high accuracy. It is an ensemble method that combines multiple decision trees to reduce overfitting and improve generalization. Random Forest can capture non-linear relationships and interactions between features, making it well-suited for diverse datasets.

Hyperparameter Tuning:

We tuned the n_estimators hyperparameter on the validation set, testing values of 50, 100, 150, and 200. The optimal number of trees was found to be 200, striking a balance between complexity and performance. The tuning process helped in identifying the right model complexity needed to capture the dataset's characteristics effectively.

Performance Improvement:

The performance improvement observed with the Random Forest model can be attributed to its ensemble nature, which aggregates predictions from multiple trees, thereby increasing the model's robustness and accuracy. Unlike the nearest neighbor approach, which relies on local data similarity, Random Forest considers a broader range of features and their interactions, leading to more precise decision-making.

```
====================Random Forest====================
Random Forest - Best Params: {'classifier__n_estimators': 200}
Random Forest - Training Accuracy: 1.0
Random Forest - Testing Accuracy: 0.9240506329113924

Testing Data:
              precision    recall  f1-score   support

           0       0.86      0.93      0.89        27
           1       0.96      0.92      0.94        52

    accuracy                           0.92        79
   macro avg       0.91      0.92      0.92        79
weighted avg       0.93      0.92      0.92        79

Confusion Matrix:
[[25  2]
 [ 4 48]]
```

Figure 4

The Random Forest model achieved a training accuracy of 100% and a testing accuracy of 92.4%. The high training accuracy might indicate overfitting; however, the model's strong testing accuracy suggests effective learning and generalization to unseen data. The precision, recall, and F1-scores further confirmed the model's balanced performance across both classes (Pass and Fail). The confusion matrix reveals a low number of misclassifications, affirming the model's robust predictive performance.

Logistic Regression Classifier

Motivation for Selection:

Logistic Regression was selected for its simplicity and interpretability. It is a linear model that works well with binary classification tasks. Logistic Regression provides insights into the relationship between features and the outcome, which is valuable for understanding the underlying patterns in the data.

Hyperparameter Tuning:

The C hyperparameter, representing the regularization strength, was tuned with values of 0.1, 1, 10, and 100 on the validation set. The best performance was achieved with a C value of 10, which helped prevent overfitting while maintaining enough model flexibility to capture the important patterns in the data.

Performance Improvement:

The improvement in performance with Logistic Regression is likely due to its linear decision boundary, which, despite its simplicity, can be quite effective in binary classification tasks. The model's ability to weigh features appropriately and its robustness to irrelevant features contributed to its improved performance over the baseline model.

```
==================== Logistic Regression ====================
Logistic Regression - Best Params: {'classifier__C': 10}
Logistic Regression - Training Accuracy: 0.9335443037974683
Logistic Regression - Testing Accuracy: 0.9113924050632911

Testing Data:
              precision    recall  f1-score   support

           0       0.83      0.93      0.88        27
           1       0.96      0.90      0.93        52

    accuracy                           0.91        79
   macro avg       0.90      0.91      0.90        79
weighted avg       0.92      0.91      0.91        79

Confusion Matrix:
[[25  2]
 [ 5 47]]
```

Figure 5

The Logistic Regression model demonstrated a training accuracy of 93.4% and a testing accuracy of 91.1%. Compared to Random Forest, Logistic Regression showed a smaller gap between training and testing accuracy, which could indicate better generalization capabilities. The

precision, recall, and F1-scores were also high, although slightly lower than those of the Random Forest model. The confusion matrix indicates a marginally higher number of misclassifications compared to Random Forest, but still denotes a high level of accuracy.

Performance Discussion:

The evaluation of the Random Forest and Logistic Regression models yielded insightful results, shedding light on each model's capabilities in classifying students as pass or fail. This performance discussion analyzes these results in detail, focusing on accuracy, precision, recall, F1-score, and the implications of these metrics.

Both Random Forest and Logistic Regression models showed significant improvements over the baseline model. The choice of hyperparameters played a crucial role in optimizing their performance. Random Forest excelled in handling the dataset's complexity, while Logistic Regression provided a simpler yet effective alternative. The differences in performance metrics between the two models can be attributed to their inherent algorithmic characteristics and the nature of the dataset.

Analysis:

In the evaluation of machine learning models for predicting student pass/fail outcomes, the Random Forest Classifier emerged as the best performer. This model was selected due to its high testing accuracy of 92.4%, robust precision and recall across both classes, and a balanced F1-score. The Random Forest model's ability to handle complex, non-linear data relationships and reduce overfitting through its ensemble approach contributed significantly to its superior performance.

In our exploration of the Random Forest model's performance, we conducted a detailed analysis of the test set instances where the model exhibited classification errors. By examining these misclassifications, we aimed to uncover any patterns or trends that could inform future improvements to the model.

The error analysis focused on six cases where the model's predictions did not align with the actual outcomes. The descriptive statistics of these error instances were as follows:

Average freetime: 3.5
Average number of absences: 5.17
Average first period grade (G1): 8.5
Average second period grade (G2): 8.67
Average number of past class failures: 1.0
Age range: 15 to 18 years

Histograms were generated to visualize the distribution of 'absences', 'failures', 'G1', and 'G2' in both error instances and the overall test set. These histograms revealed that:

Absences: Misclassified instances were not heavily skewed towards higher absence counts, indicating that the number of absences alone may not be a key differentiator for the model.

Failures: A higher mean of past class failures was observed in error instances compared to the overall test set, suggesting that the model might struggle with students who have a history of failures.

Grades: Lower average grades (G1 and G2) in error instances point to potential difficulties the model faces with students on the borderline of passing and failing.

Interestingly, the patterns observed in 'failures', 'G1', and 'G2' suggest that academic performance is a significant factor in the model's prediction errors. Students with lower grades and more failures were more likely to be misclassified. This insight is particularly important as it directs attention to the academic features as areas for model refinement.

The age of students did not appear to be a distinguishing factor in misclassifications, as the age range was relatively consistent with the general test set population. However, it is important to consider that the small sample size of error instances might limit the generalizability of these findings.

The performance analysis revealed that while the Random Forest model is highly accurate overall, it is more prone to making errors with students who have lower academic performance and a history of failures. These findings underscore the potential need to further investigate the influence of academic-related features and to consider additional data or feature techniques that might improve the model's ability to classify these particular instances correctly.

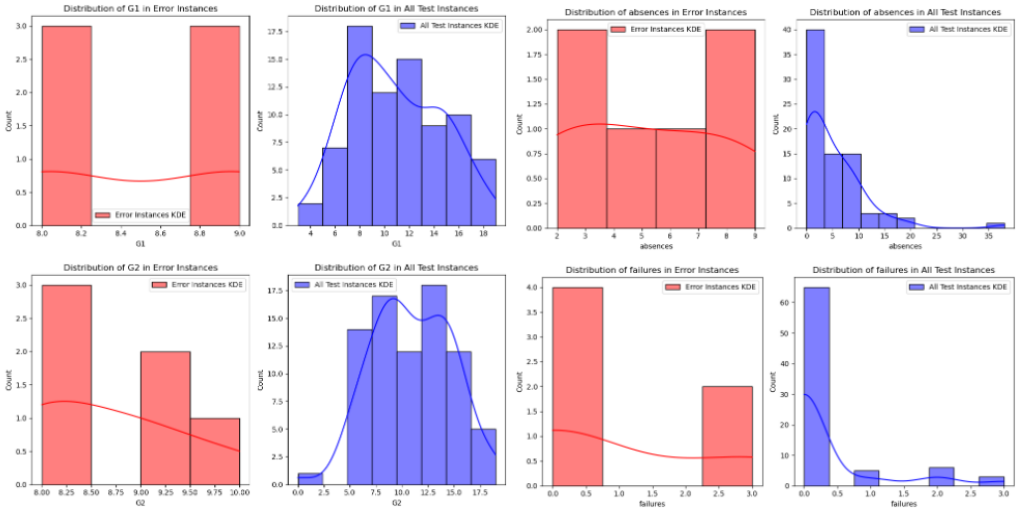

Figure 6

errors instances:

	freetime	Fedu	Medu	Walc	goout	age	failures	absences	G1	G2	\
78	5	1	2	1	1	17	3	2	8	8	
375	3	1	1	2	2	18	0	2	8	8	
377	4	4	4	4	3	18	0	4	8	9	
157	2	1	1	5	5	18	3	6	9	8	
317	3	3	4	1	4	18	0	9	9	10	
114	4	1	2	1	2	15	0	8	9	9	

	Actual	Predicted	Error
78	1	0	True
375	1	0	True
377	1	0	True
157	1	0	True
317	0	1	True
114	0	1	True

Error Instances Analysis:

	freetime	Fedu	Medu	Walc	goout	age	failures	\
count	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	
mean	3.500000	1.833333	2.333333	2.333333	2.833333	17.333333	1.000000	
std	1.048809	1.329160	1.366260	1.751190	1.471960	1.211060	1.549193	
min	2.000000	1.000000	1.000000	1.000000	1.000000	15.000000	0.000000	
25%	3.000000	1.000000	1.250000	1.000000	2.000000	17.250000	0.000000	
50%	3.500000	1.000000	2.000000	1.500000	2.500000	18.000000	0.000000	
75%	4.000000	2.500000	3.500000	3.500000	3.750000	18.000000	2.250000	
max	5.000000	4.000000	4.000000	5.000000	5.000000	18.000000	3.000000	

	absences	G1	G2	Actual	Predicted
count	6.000000	6.000000	6.000000	6.000000	6.000000
mean	5.166667	8.500000	8.666667	0.666667	0.333333
std	2.994439	0.547723	0.816497	0.516398	0.516398
min	2.000000	8.000000	8.000000	0.000000	0.000000
25%	2.500000	8.000000	8.000000	0.250000	0.000000
50%	5.000000	8.500000	8.500000	1.000000	0.000000
75%	7.500000	9.000000	9.000000	1.000000	0.750000
max	9.000000	9.000000	10.000000	1.000000	1.000000

Figure 7

Conclusion and Discussion:

In our investigation into student performance prediction, the Random Forest Classifier emerged as the standout model, delivering high accuracy and well-balanced evaluation metrics. Its ensemble approach proved adept at navigating the complexities of the data, showcasing the power of aggregating multiple decision trees to improve prediction robustness.

However, no model is without its limitations. The K-Nearest Neighbors algorithm demonstrated sensitivity to data noise, potentially affecting its reliability. Random Forest, while robust, faced the risk of overfitting due to its complexity. Logistic Regression, with its assumption of linearity, might not fully capture the intricacies of the data. Insights from our error analysis suggested that targeted feature refinement and the adoption of more advanced modeling techniques could enhance model performance. Additionally, the study's scope was naturally limited by the dataset's size and the inherent biases present within the models themselves.

Despite these constraints, our research lays the groundwork for future exploration. We underscore the importance of a comprehensive evaluation strategy that goes beyond accuracy to include precision, recall, and other relevant metrics, particularly within real-world educational contexts. Looking ahead, there is a compelling opportunity to leverage larger, more varied datasets and to employ more advanced modeling strategies. Such advancements hold promise for strengthening predictive accuracy and offering more designed support to students who may be on the edge of academic difficulties.