GPA Genie:

Integrating Study Habits, Extracurricular Activities, and Parental Information for GPA Classification

Dhruv Chandna & Soham Jain

Thomas Jefferson High School for Science and Technology

TJ Machine Learning 1

Dr. Yilmaz

October 21, 2024

Table of Contents

Part 1 - Project Overview

Every year, the stresses placed upon young shoulders become heavier. As time for college applications approach, students lean into cycles of self-doubt and regret, wondering if their in and out of school activities are sufficient to receive an admission letter from the Harvards and Stanfords of the world.

Imagine a world where students could control their own destinies. GPA Genie serves to act as a tool with which students can optimize their extracurricular activities from an early age, thereby helping to secure later success. Students will be able to use the tool to weigh different configurations of academic options and decide which ones make most sense for them.

In this project, our goal is to use the Students Performance Dataset to classify a predicted GPA range given a certain quantitative configuration of study habits, extracurricular, and parental involvement. The class we are predicting is called *GradeClass*, which ascribes the labels *0, 1, 2, 3, 4* to GPAs ranging $GPA \geq 3.5$, $3.0 \leq GPA < 3.5$, $2.5 \leq GPA < 3.0$, $2.0 \leq GPA < 2.5$, and $GPA < 2$, respectively.

Part 2 - Dataset

Link to dataset: https://www.kaggle.com/datasets/rabieelkharoua/students-performance-dataset

The *Students Performance Dataset* contains detailed information about 2,392 high school students, including their demographics, study habits, parental involvement, extracurricular activities, and academic performance. The target variable, GradeClass, categorizes students' grades into distinct groups, making it a valuable resource for educational research.

The first column corresponds to the student's identification number (Student ID), which is randomly assigned from 1001 to 3392. Following this column, there are attributes describing the students' academic profiles.

The attributes are defined as follows:

1. Age — 15 to 18
2. Gender — 0 to 1
    a. 0: Male
    b. 1: Female
3. Ethnicity — 0 to 3
    a. 0: Caucasian
    b. 1: African American
    c. 2: Asian
    d. 3: Other
4. ParentalEducation — 0 to 4
    a. 0: None
    b. 1: High School
    c. 2: Some College

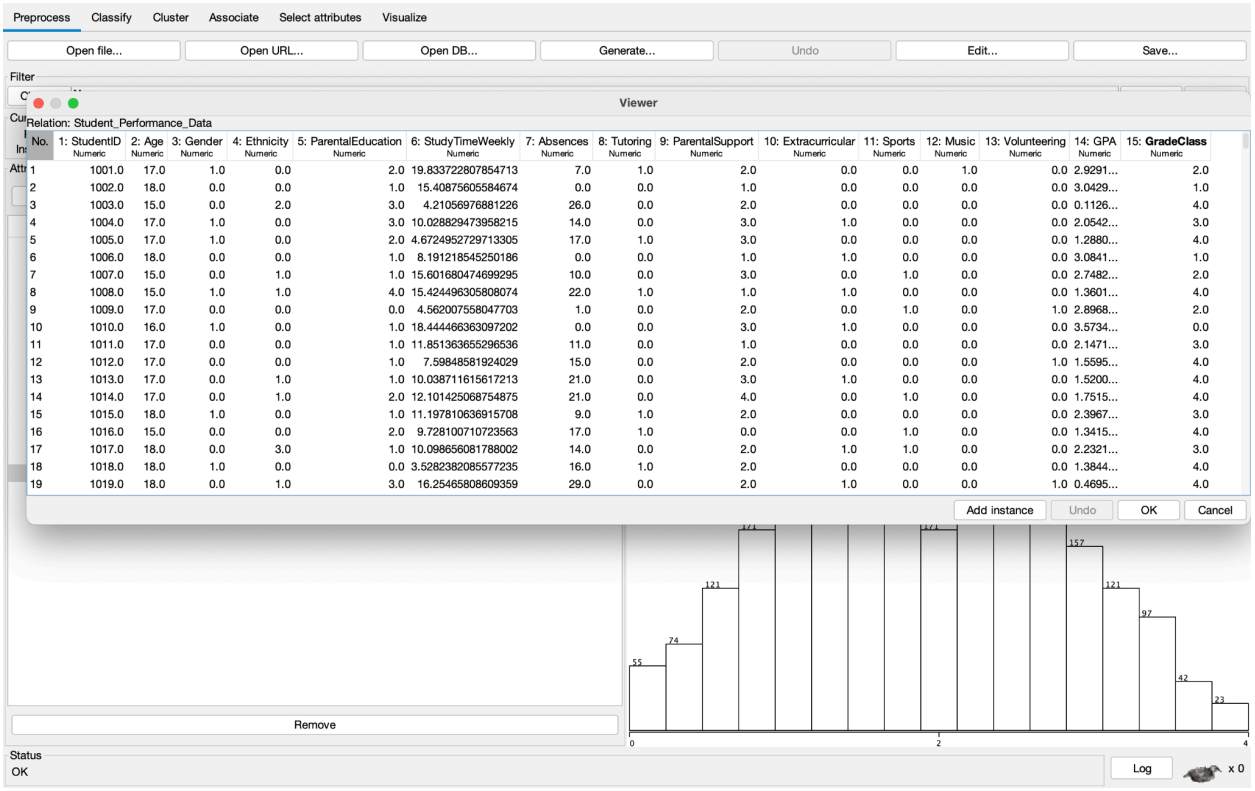
    d. 3: Bachelor's
    e. 4: Higher
5. StudyTimeWeekly — 0.0 to 20.0
    a. Weekly study time in hours as a quantitative continuous variable
6. Absences — 0 to 30
    a. Number of absences during the school year as a quantitative discrete variable
7. Tutoring — 0 or 1
    a. Tutoring status, where 0 indicates No and 1 indicates Yes
8. ParentalSupport — 0 to 4
    a. Self-evaluated by the student
    b. 0: None
    c. 1: Low
    d. 2: Moderate
    e. 3: High
    f. 4: Very High
9. Extracurricular — 0 or 1
    a. Participation in extracurricular activities
    b. 0: No
    c. 1: Yes
10. Sports — 0 or 1
    a. Participation in sports
    b. 0: No
    c. 1: Yes
11. Music — 0 or 1
    a. Participation in music activities
    b. 0: No
    c. 1: Yes
12. Volunteering — 0 or 1
    a. Participation in volunteering
    b. 0: No
    c. 1: Yes
13. GPA — 2.0 to 4.0
    a. Grade Point Average on a scale from 2.0 to 4.0

The GPA attribute is a quantitative continuous value that was rounded to generate the GradeClass values. Therefore, we removed this attribute since the class itself is derived from this attribute. Altogether, the dataset contains 12 attributes, and it has a dimension of 12 as well. The dataset also contains 2,392 instances, each of which represents a high-school student. There are no missing values in the dataset. Most of the attributes are uniformly distributed, with the exception of right-skewed data for Ethnicity, Tutoring, Extracurricular, Sports, Music, and Volunteering.

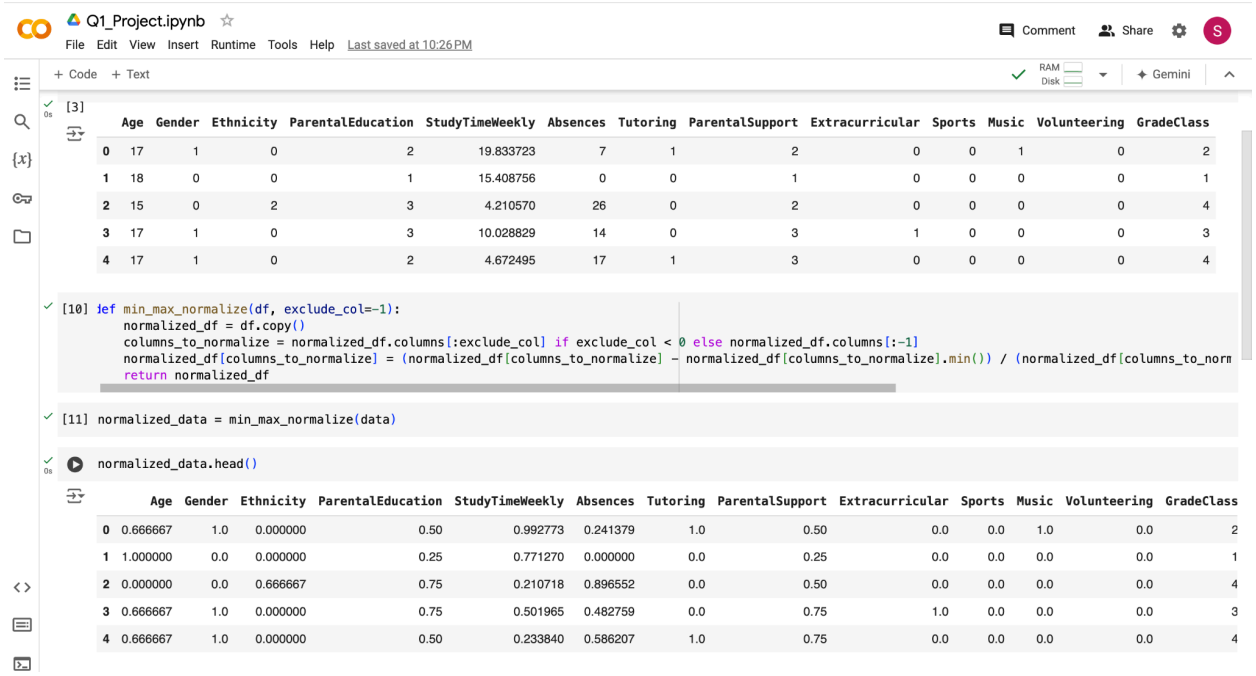
Part 3 - Preprocessing

Typically, we would begin by removing any instances with empty class labels as a supervised learning task requires labels. However, since the dataset does not contain any missing values, there is no need to remove any instances or attributes.

We first loaded the dataset by downloading it as a CSV from Kaggle and uploading it onto WEKA. The figure below shows the dataset when it was first uploaded:



Next, we ran min-max normalization to ensure that all of the values range between 0 and 1. In order to do this, we transferred our dataset as a CSV over to Google Colab. The screenshot below shows our dataset before and after normalization:



Q1_Project.ipynb

File Edit View Insert Runtime Tools Help Last saved at 10:26 PM

Comment Share

+ Code + Text

RAM Disk Gemini

[3]

	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GradeClass
0	17	1	0	2	19.833723	7	1	2	0	0	1	0	2
1	18	0	0	1	15.408756	0	0	1	0	0	0	0	1
2	15	0	2	3	4.210570	26	0	2	0	0	0	0	4
3	17	1	0	3	10.028829	14	0	3	1	0	0	0	3
4	17	1	0	2	4.672495	17	1	3	0	0	0	0	4

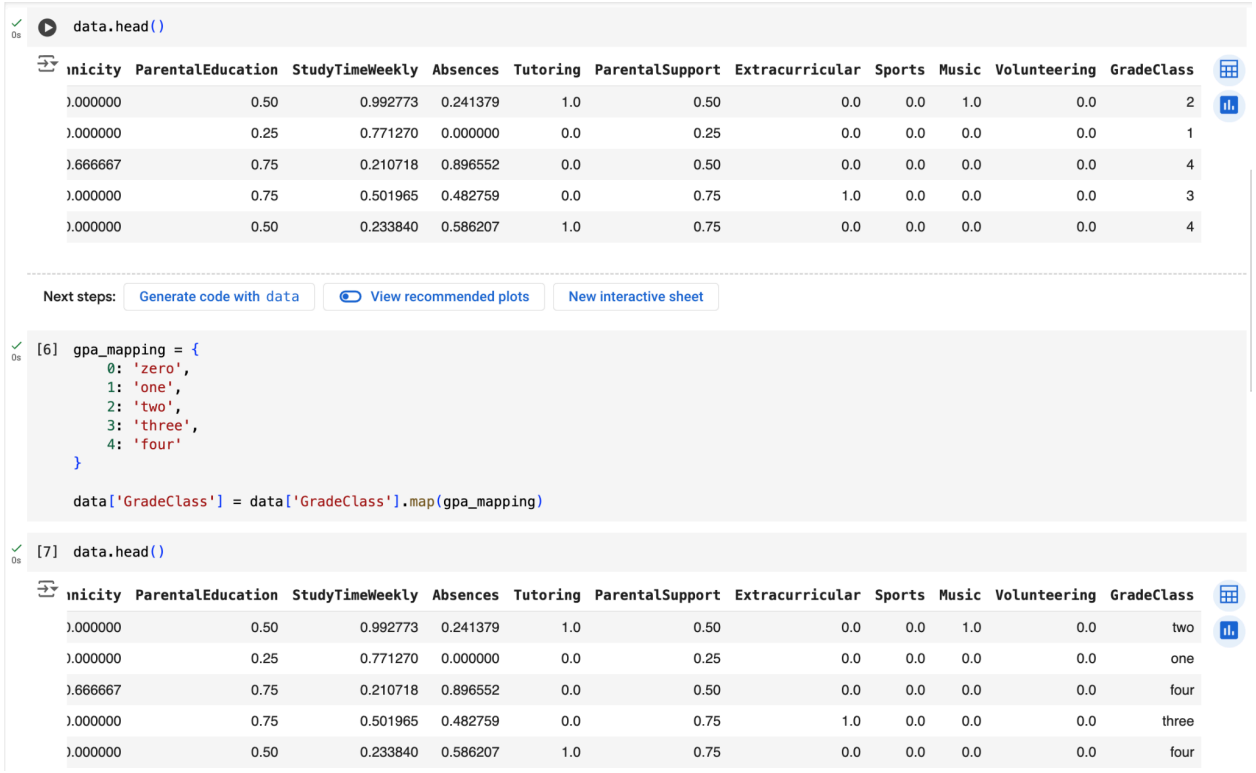
[10]
```
def min_max_normalize(df, exclude_col=-1):
    normalized_df = df.copy()
    columns_to_normalize = normalized_df.columns[:exclude_col] if exclude_col < 0 else normalized_df.columns[:-1]
    normalized_df[columns_to_normalize] = (normalized_df[columns_to_normalize] - normalized_df[columns_to_normalize].min()) / (normalized_df[columns_to_norm
    return normalized_df
```

[11]
```
normalized_data = min_max_normalize(data)
```

```
normalized_data.head()
```

	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GradeClass
0	0.666667	1.0	0.000000	0.50	0.992773	0.241379	1.0	0.50	0.0	0.0	1.0	0.0	2
1	1.000000	0.0	0.000000	0.25	0.771270	0.000000	0.0	0.25	0.0	0.0	0.0	0.0	1
2	0.000000	0.0	0.666667	0.75	0.210718	0.896552	0.0	0.50	0.0	0.0	0.0	0.0	4
3	0.666667	1.0	0.000000	0.75	0.501965	0.482759	0.0	0.75	1.0	0.0	0.0	0.0	3
4	0.666667	1.0	0.000000	0.50	0.233840	0.586207	1.0	0.75	0.0	0.0	0.0	0.0	4

After normalizing the dataset, we alphabetized the class values by converting them from quantitative to qualitative data for the purpose of classification. For instance, a GradeClass value of '1' was converted to 'one', '2' was converted to 'two', and so forth. We did this through mapping, as shown in the screenshot below:



```
data.head()
```

nicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GradeClass
.000000	0.50	0.992773	0.241379	1.0	0.50	0.0	0.0	1.0	0.0	2
.000000	0.25	0.771270	0.000000	0.0	0.25	0.0	0.0	0.0	0.0	1
.666667	0.75	0.210718	0.896552	0.0	0.50	0.0	0.0	0.0	0.0	4
.000000	0.75	0.501965	0.482759	0.0	0.75	1.0	0.0	0.0	0.0	3
.000000	0.50	0.233840	0.586207	1.0	0.75	0.0	0.0	0.0	0.0	4

Next steps: Generate code with data View recommended plots New interactive sheet

[6]
```
gpa_mapping = {
    0: 'zero',
    1: 'one',
    2: 'two',
    3: 'three',
    4: 'four'
}

data['GradeClass'] = data['GradeClass'].map(gpa_mapping)
```

[7]
```
data.head()
```

nicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GradeClass
.000000	0.50	0.992773	0.241379	1.0	0.50	0.0	0.0	1.0	0.0	two
.000000	0.25	0.771270	0.000000	0.0	0.25	0.0	0.0	0.0	0.0	one
.666667	0.75	0.210718	0.896552	0.0	0.50	0.0	0.0	0.0	0.0	four
.000000	0.75	0.501965	0.482759	0.0	0.75	1.0	0.0	0.0	0.0	three
.000000	0.50	0.233840	0.586207	1.0	0.75	0.0	0.0	0.0	0.0	four

We downloaded this dataset as a CSV from Google Colab and uploaded it into our folder.

Part 4 - Attribute Selection

Class Attribute: **GradeClass**
Features (12): **Age, Gender, Ethnicity, ParentalEducation, StudyTimesWeekly, Absences, Tutoring, ParentalSupport, Extracurricular, Sports, Music, Volunteering**

Method 1: ***Ranker + CorrelationAttributeEval***

CorrelationAttributeEval is an attribute selection algorithm that evaluates attributes by measuring Pearson's correlation coefficient. The equation for Pearson's correlation coefficient is given by:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$, where each of the variables is defined as:

- r = correlation coefficient
- x_i = individual values of x
- $\bar{x}$ = mean (average) of x
- y_i = individual values of y
- $\bar{y}$ = mean (average) of y

After evaluating our dataset on this attribute selection algorithm, we achieved the following results:

```
Ranked attributes:
 0.017224  11 Music
 0.013468   3 Ethnicity
 0.013074   1 Age
 0.01196    4 ParentalEducation
-0.0002     2 Gender
-0.002799  10 Sports
-0.006036   8 ParentalSupport
-0.007427   9 Extracurricular
-0.016604   5 StudyTimeWeekly
-0.018528   6 Absences
-0.050898   7 Tutoring
```

Using a threshold of 0.01, we find that we must:

- *Remove:*
  - **Gender**
  - **Sports**
  - **ParentalSupport**
  - **Extracurricular**
  - **Volunteering**
- *Retain:*
  - **Music**
  - **Ethnicity**
  - **Age**
  - **ParentalEducation**
  - **StudyTimeWeekly**
  - **Absences**
  - **Tutoring**

Method 2: ***Ranker + ReliefFAttributeEval***

ReliefFAttributeEval evaluates how well each attribute distinguishes between instances of different classes based on local neighborhoods in the feature space, allowing for effective feature ranking and selection. The results from using this algorithm in Weka are shown below:

```
Ranked attributes:
 0.005512094   6 Absences
 0.002713589   9 Extracurricular
 0.001505272  11 Music
 0.00049426   10 Sports
 0.000000356   2 Gender
-0.000436036   7 Tutoring
-0.001040747   8 ParentalSupport
-0.001403292   4 ParentalEducation
-0.006471977   3 Ethnicity
-0.006616634   1 Age
-0.006968669   5 StudyTimeWeekly
```

Using Ranker + ReliefFAttributeEval with a threshold of 0.0015, we find that we:

- *Remove*
  - **Sports**
  - **Gender**
  - **Tutoring**
  - **ParentalSupport**
  - **ParentalEducation**
  - **Volunteering**
- *Retain*
  - **Absences**
  - **Extracurricular**
  - **Music**
  - **Ethnicity**
  - **Age**
  - **StudyTimeWeekly**

Method 3: ***GreedyStepwise + CfsSubsetEval***

CfsSubsetEval considers the ability of each attribute to predict the class values by evaluating its relevance while taking into account the redundancy among attributes. It identifies subsets of attributes that work well together, enuring that the selected features provide the best predictive power without significant overlap. Our evaluation on Weka is shown below:

```
Selected attributes: 1,3,4,5,6,7,11 : 7
                     Age
                     Ethnicity
                     ParentalEducation
                     StudyTimeWeekly
                     Absences
                     Tutoring
                     Music
```

Using GreedyStepwise CfsSubsetEval, we find that we:

- *Remove*
  - **Gender**
  - **ParentalSupport**
  - **Extracurricular**
  - **Sports**
  - **Volunteering**
- *Retain*
  - **Age**
  - **Ethnicity**
  - **ParentalEducation**
  - **StudyTimeWeekly**
  - **Absences**
  - **Tutoring**
  - **Music**

Method 4: ***Ranker + PrincipalComponents***

Principal component analysis, or PCA, calculates the eigenvectors and eigenvalues of the covariance matrix of the original attributes, identifying the directions that maximize variance. By selecting the top principal components, PCA reduces the dimensionality of the dataset while retaining the most informative features. Our results from Weka are shown below:

```
Ranked attributes:
 0.9067   1 -0.451Age+0.415Tutoring-0.365Gender-0.339Volunteering-0.31Music...
 0.8168   2 -0.482ParentalSupport-0.478StudyTimeWeekly-0.368Tutoring+0.305ParentalEducation+0.295Volunteering...
 0.728    3 -0.643Absences-0.471Sports-0.327StudyTimeWeekly-0.299ParentalEducation-0.246Gender...
 0.6412   4 0.64 Ethnicity-0.461Age-0.298Absences+0.242Volunteering-0.241Gender...
 0.556    5 0.656Music+0.432ParentalEducation-0.368Gender-0.26Ethnicity+0.258Extracurricular...
 0.4716   6 0.497Ethnicity+0.482Extracurricular-0.42Volunteering+0.314ParentalEducation-0.285Sports...
 0.389    7 -0.62Extracurricular-0.464ParentalSupport+0.404ParentalEducation+0.334Tutoring+0.225Age...
 0.3083   8 0.54 ParentalSupport-0.497Gender+0.37 Age-0.338Music-0.325StudyTimeWeekly...
 0.2283   9 0.598StudyTimeWeekly+0.566Volunteering+0.33 Extracurricular-0.258Music+0.18 Age...
 0.1494  10 0.578Sports+0.426Tutoring+0.424Gender-0.37Absences-0.228StudyTimeWeekly...
 0.0727  11 0.518Tutoring+0.474Absences+0.396Volunteering-0.34Sports-0.293StudyTimeWeekly...
 0       12 0.474Age-0.425ParentalEducation+0.415Ethnicity+0.363Music-0.34ParentalSupport...
```

Using a cutoff value of 0.7, the attributes that we retain are:

- -0.451Age+0.415Tutoring-0.365Gender-0.339Volunteering-0.31Music...
- -0.482ParentalSupport-0.478StudyTimeWeekly-0.368Tutoring+0.305ParentalEducation+0.295Volunteering...
- -0.643Absences-0.471Sports-0.327StudyTimeWeekly-0.299ParentalEducation-0.246Gender...

*Method 5: **AllRetained***

For our fifth attribute selection approach, we did not remove any attributes. This is because we want to see if retaining all the attributes optimizes the overall performance of the model, since some attributes may rely on others to capture trends in the class values.

Part 5 - Train-Validation-Test Split

For each of our five datasets, which were generated through the five methods from attribute selection, we created train, test, and validation datasets with a split of 80%, 10%, and 10%, respectively. We applied a stratified split by utilizing the train_test_split method from scikit-learn and the *stratify* parameter. The screenshot below shows the split_df method:

```
def split_df(df, target_column, train_split=0.8, test_split=0.1, val_split=0.1):
    train_df, temp_df = train_test_split(df, test_size=(1 - train_split), stratify=df[target_column], random_state=42)
    test_df, val_df = train_test_split(temp_df, test_size=(val_split / (test_split + val_split)), stratify=temp_df[target_column], random_state=42)

    return train_df, test_df, val_df
```

Using this method, we applied the train-validation-test split to each of the datasets. The code below shows how we used mapping and split_df while iterating through the datasets:

```
num_to_word = {0: 'zero', 1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
for folder in folders:
    subdataset_folder_path = os.path.join(dataset_folder_path, folder)
    files = os.listdir(subdataset_folder_path)
    file = [file for file in files if file == f"{folder}_Dataset.csv"][0]
    file_path = os.path.join(subdataset_folder_path, file)
    df = pd.read_csv(file_path)
    df["GradeClass"] = df["GradeClass"].map(num_to_word)
    train_df, test_df, val_df = split_df(df, "GradeClass")
    train_df.to_csv(os.path.join(subdataset_folder_path, f"{folder}_train.csv"), index=False)
    test_df.to_csv(os.path.join(subdataset_folder_path, f"{folder}_test.csv"), index=False)
    val_df.to_csv(os.path.join(subdataset_folder_path, f"{folder}_val.csv"), index=False)
```

Then, we saved the datasets as CSVs. Since the original dataset contained 2392 instances, the train, test, and validation datasets consist of 1913, 240, and 239 attributes, respectively.

Part 6 - Classifiers

We evaluated each dataset on four model classifiers through Weka: Logistic, MultilayerPerceptron, Bagging, and LMT.

CorrelationAttributeEval with Logistic

```
=== Summary ===

Correctly Classified Instances         171               71.5481 %
Incorrectly Classified Instances        68               28.4519 %
Kappa statistic                          0.5608
Mean absolute error                      0.1716
Root mean squared error                  0.2857
Relative absolute error                 63.7603 %
Root relative squared error             77.8761 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.161    0.858      0.950    0.902      0.795    0.951     0.929     four
                 0.390    0.076    0.516      0.390    0.444      0.353    0.844     0.532     three
                 0.000    0.000    ?          0.000    ?          ?        0.803     0.281     zero
                 0.519    0.033    0.667      0.519    0.583      0.543    0.872     0.581     one
                 0.667    0.135    0.491      0.667    0.565      0.473    0.878     0.510     two
Weighted Avg.    0.715    0.120    ?          0.715    ?          ?        0.905     0.723

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 115   5   0   0   1 |   a = four
  11  16   0   0  14 |   b = three
   3   0   0   3   5 |   c = zero
   3   3   0  14   7 |   d = one
   2   7   0   4  26 |   e = two
```

CorrelationAttributeEval with MultilayerPerceptron

```
=== Summary ===

Correctly Classified Instances         165               69.0377 %
Incorrectly Classified Instances        74               30.9623 %
Kappa statistic                          0.5397
Mean absolute error                      0.1499
Root mean squared error                  0.2845
Relative absolute error                 55.6973 %
Root relative squared error             77.5701 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.884    0.085    0.915      0.884    0.899      0.800    0.944     0.926     four
                 0.585    0.136    0.471      0.585    0.522      0.413    0.829     0.484     three
                 0.091    0.018    0.200      0.091    0.125      0.107    0.816     0.289     zero
                 0.296    0.033    0.533      0.296    0.381      0.344    0.859     0.481     one
                 0.641    0.130    0.490      0.641    0.556      0.461    0.874     0.527     two
Weighted Avg.    0.690    0.092    0.693      0.690    0.684      0.595    0.898     0.705

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 107  13   0   0   1 |   a = four
   4  24   0   0  13 |   b = three
   2   1   1   4   3 |   c = zero
   3   3   4   8   9 |   d = one
   1  10   0   3  25 |   e = two
```

CorrelationAttributeEval with Bagging

```
=== Summary ===

Correctly Classified Instances         161               67.364  %
Incorrectly Classified Instances        78               32.636  %
Kappa statistic                          0.5043
Mean absolute error                      0.16
Root mean squared error                  0.2915
Relative absolute error                 59.4337 %
Root relative squared error             79.4648 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.926    0.136    0.875      0.926    0.900      0.792    0.943     0.919     four
                 0.512    0.116    0.477      0.512    0.494      0.385    0.828     0.464     three
                 0.091    0.018    0.200      0.091    0.125      0.107    0.700     0.160     zero
                 0.296    0.061    0.381      0.296    0.333      0.263    0.803     0.393     one
                 0.487    0.110    0.463      0.487    0.475      0.370    0.853     0.436     two
Weighted Avg.    0.674    0.114    0.653      0.674    0.661      0.562    0.882     0.668

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 112   7   0   0   2 |   a = four
  10  21   0   1   9 |   b = three
   1   2   1   5   2 |   c = zero
   4   2   4   8   9 |   d = one
   1  12   0   7  19 |   e = two
```

CorrelationAttributeEval with LMT

```
=== Summary ===

Correctly Classified Instances         174               72.8033 %
Incorrectly Classified Instances        65               27.1967 %
Kappa statistic                          0.5843
Mean absolute error                      0.1572
Root mean squared error                  0.2774
Relative absolute error                 58.4123 %
Root relative squared error             75.6121 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.136    0.877      0.942    0.908      0.810    0.951     0.934     four
                 0.561    0.091    0.561      0.561    0.561      0.470    0.853     0.546     three
                 0.000    0.000    ?          0.000    ?          ?        0.770     0.231     zero
                 0.519    0.057    0.538      0.519    0.528      0.470    0.857     0.507     one
                 0.590    0.095    0.548      0.590    0.568      0.480    0.873     0.606     two
Weighted Avg.    0.728    0.106    ?          0.728    ?          ?        0.903     0.733

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 114   6   0   0   1 |   a = four
   9  23   0   0   9 |   b = three
   2   1   0   6   2 |   c = zero
   3   3   0  14   7 |   d = one
   2   8   0   6  23 |   e = two
```

ReliefFAttributeEval with Logistic

```
=== Summary ===

Correctly Classified Instances         171               71.5481 %
Incorrectly Classified Instances        68               28.4519 %
Kappa statistic                          0.5572
Mean absolute error                      0.1727
Root mean squared error                  0.2873
Relative absolute error                 64.1416 %
Root relative squared error             78.325  %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.959    0.178    0.847      0.959    0.899      0.789    0.950     0.922     four
                 0.415    0.056    0.607      0.415    0.493      0.421    0.858     0.577     three
                 0.000    0.000    ?          0.000    ?          ?        0.826     0.383     zero
                 0.296    0.038    0.500      0.296    0.372      0.327    0.857     0.535     one
                 0.769    0.140    0.517      0.769    0.619      0.542    0.879     0.507     two
Weighted Avg.    0.715    0.127    ?          0.715    ?          ?        0.906     0.727

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 116   4   0   0   1 |   a = four
  13  17   0   0  11 |   b = three
   3   0   0   6   2 |   c = zero
   3   2   0   8  14 |   d = one
   2   5   0   2  30 |   e = two
```

ReliefFAttributeEval with MultilayerPerceptron

```
=== Summary ===

Correctly Classified Instances         171               71.5481 %
Incorrectly Classified Instances        68               28.4519 %
Kappa statistic                          0.5702
Mean absolute error                      0.1522
Root mean squared error                  0.2783
Relative absolute error                 56.525  %
Root relative squared error             75.8689 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.926    0.102    0.903      0.926    0.914      0.824    0.951     0.927     four
                 0.488    0.091    0.526      0.488    0.506      0.409    0.856     0.540     three
                 0.000    0.009    0.000      0.000    0.000      -0.020   0.823     0.248     zero
                 0.333    0.038    0.529      0.333    0.409      0.364    0.850     0.444     one
                 0.769    0.140    0.517      0.769    0.619      0.542    0.898     0.580     two
Weighted Avg.    0.715    0.095    0.692      0.715    0.697      0.616    0.909     0.718

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 112   8   0   0   1 |   a = four
   7  20   0   0  14 |   b = three
   1   2   0   6   2 |   c = zero
   3   2   2   9  11 |   d = one
   1   6   0   2  30 |   e = two
```

ReliefFAttributeEval with Bagging

```
=== Summary ===

Correctly Classified Instances         170               71.1297 %
Incorrectly Classified Instances        69               28.8703 %
Kappa statistic                          0.5611
Mean absolute error                      0.1569
Root mean squared error                  0.2834
Relative absolute error                 58.2708 %
Root relative squared error             77.2543 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.926    0.136    0.875      0.926    0.900      0.792    0.946     0.909     four
                 0.537    0.086    0.564      0.537    0.550      0.460    0.830     0.476     three
                 0.000    0.009    0.000      0.000    0.000      -0.020   0.795     0.228     zero
                 0.407    0.061    0.458      0.407    0.431      0.365    0.810     0.405     one
                 0.641    0.105    0.543      0.641    0.588      0.502    0.880     0.550     two
Weighted Avg.    0.711    0.108    0.680      0.711    0.695      0.602    0.893     0.688

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 112   6   0   0   3 |   a = four
  11  22   0   1   7 |   b = three
   1   2   0   6   2 |   c = zero
   3   2   2  11   9 |   d = one
   1   7   0   6  25 |   e = two
```

ReliefFAttributeEval with LMT

```
=== Summary ===

Correctly Classified Instances         173               72.3849 %
Incorrectly Classified Instances        66               27.6151 %
Kappa statistic                          0.5801
Mean absolute error                      0.1574
Root mean squared error                  0.2776
Relative absolute error                 58.4865 %
Root relative squared error             75.6858 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.110    0.898      0.942    0.919      0.834    0.947     0.916     four
                 0.585    0.096    0.558      0.585    0.571      0.480    0.850     0.504     three
                 0.000    0.000    ?          0.000    ?          ?        0.778     0.283     zero
                 0.370    0.057    0.455      0.370    0.408      0.344    0.811     0.467     one
                 0.641    0.110    0.532      0.641    0.581      0.494    0.896     0.602     two
Weighted Avg.    0.724    0.097    ?          0.724    ?          ?        0.899     0.714

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 114   6   0   0   1 |   a = four
   8  24   0   0   9 |   b = three
   1   2   0   7   1 |   c = zero
   3   3   0  10  11 |   d = one
   1   8   0   5  25 |   e = two
```

CfsSubsetEval with Logistic

```
=== Summary ===

Correctly Classified Instances         171               71.5481 %
Incorrectly Classified Instances        68               28.4519 %
Kappa statistic                          0.5608
Mean absolute error                      0.1716
Root mean squared error                  0.2857
Relative absolute error                 63.7603 %
Root relative squared error             77.8761 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.161    0.858      0.950    0.902      0.795    0.951     0.929     four
                 0.390    0.076    0.516      0.390    0.444      0.353    0.844     0.532     three
                 0.000    0.000    ?          0.000    ?          ?        0.803     0.281     zero
                 0.519    0.033    0.667      0.519    0.583      0.543    0.872     0.581     one
                 0.667    0.135    0.491      0.667    0.565      0.473    0.878     0.510     two
Weighted Avg.    0.715    0.120    ?          0.715    ?          ?        0.905     0.723

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 115   5   0   0   1 |   a = four
  11  16   0   0  14 |   b = three
   3   0   0   3   5 |   c = zero
   3   3   0  14   7 |   d = one
   2   7   0   4  26 |   e = two
```

CfsSubsetEval with MultilayerPerceptron

```
=== Summary ===

Correctly Classified Instances         165               69.0377 %
Incorrectly Classified Instances        74               30.9623 %
Kappa statistic                          0.5397
Mean absolute error                      0.1499
Root mean squared error                  0.2845
Relative absolute error                 55.6973 %
Root relative squared error             77.5701 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.884    0.085    0.915      0.884    0.899      0.800    0.944     0.926     four
                 0.585    0.136    0.471      0.585    0.522      0.413    0.829     0.484     three
                 0.091    0.018    0.200      0.091    0.125      0.107    0.816     0.289     zero
                 0.296    0.033    0.533      0.296    0.381      0.344    0.859     0.481     one
                 0.641    0.130    0.490      0.641    0.556      0.461    0.874     0.527     two
Weighted Avg.    0.690    0.092    0.693      0.690    0.684      0.595    0.898     0.705

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 107  13   0   0   1 |   a = four
   4  24   0   0  13 |   b = three
   2   1   1   4   3 |   c = zero
   3   3   4   8   9 |   d = one
   1  10   0   3  25 |   e = two
```

CfsSubsetEval with Bagging

```
=== Summary ===

Correctly Classified Instances         161               67.364  %
Incorrectly Classified Instances        78               32.636  %
Kappa statistic                          0.5043
Mean absolute error                      0.16
Root mean squared error                  0.2915
Relative absolute error                 59.4337 %
Root relative squared error             79.4648 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.926    0.136    0.875      0.926    0.900      0.792    0.943     0.919     four
                 0.512    0.116    0.477      0.512    0.494      0.385    0.828     0.464     three
                 0.091    0.018    0.200      0.091    0.125      0.107    0.700     0.160     zero
                 0.296    0.061    0.381      0.296    0.333      0.263    0.803     0.393     one
                 0.487    0.110    0.463      0.487    0.475      0.370    0.853     0.436     two
Weighted Avg.    0.674    0.114    0.653      0.674    0.661      0.562    0.882     0.668

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 112   7   0   0   2 |   a = four
  10  21   0   1   9 |   b = three
   1   2   1   5   2 |   c = zero
   4   2   4   8   9 |   d = one
   1  12   0   7  19 |   e = two
```

CfsSubsetEval with LMT

```
=== Summary ===

Correctly Classified Instances         174               72.8033 %
Incorrectly Classified Instances        65               27.1967 %
Kappa statistic                          0.5843
Mean absolute error                      0.1572
Root mean squared error                  0.2774
Relative absolute error                 58.4123 %
Root relative squared error             75.6121 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.136    0.877      0.942    0.908      0.810    0.951     0.934     four
                 0.561    0.091    0.561      0.561    0.561      0.470    0.853     0.546     three
                 0.000    0.000    ?          0.000    ?          ?        0.770     0.231     zero
                 0.519    0.057    0.538      0.519    0.528      0.470    0.857     0.507     one
                 0.590    0.095    0.548      0.590    0.568      0.480    0.873     0.606     two
Weighted Avg.    0.728    0.106    ?          0.728    ?          ?        0.903     0.733

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 114   6   0   0   1 |   a = four
   9  23   0   0   9 |   b = three
   2   1   0   6   2 |   c = zero
   3   3   0  14   7 |   d = one
   2   8   0   6  23 |   e = two
```

PrincipalComponents with Logistic

```
=== Summary ===

Correctly Classified Instances         140               58.5774 %
Incorrectly Classified Instances        99               41.4226 %
Kappa statistic                          0.2622
Mean absolute error                      0.2272
Root mean squared error                  0.3377
Relative absolute error                 84.4004 %
Root relative squared error             92.0582 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.983    0.627    0.617      0.983    0.758      0.452    0.816     0.801     four
                 0.000    0.000    ?          0.000    ?          ?        0.685     0.299     three
                 0.000    0.000    ?          0.000    ?          ?        0.652     0.085     zero
                 0.148    0.019    0.500      0.148    0.229      0.228    0.753     0.419     one
                 0.436    0.105    0.447      0.436    0.442      0.334    0.712     0.318     two
Weighted Avg.    0.586    0.337    ?          0.586    ?          ?        0.762     0.560

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 119   0   0   0   2 |   a = four
  32   0   0   2   7 |   b = three
   7   0   0   1   3 |   c = zero
  14   0   0   4   9 |   d = one
  21   0   0   1  17 |   e = two
```

PrincipalComponents with MultilayerPerceptron

```
=== Summary ===

Correctly Classified Instances         144               60.251  %
Incorrectly Classified Instances        95               39.749  %
Kappa statistic                          0.3165
Mean absolute error                      0.2257
Root mean squared error                  0.3341
Relative absolute error                 83.8419 %
Root relative squared error             91.0952 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.967    0.525    0.654      0.967    0.780      0.509    0.815     0.803     four
                 0.000    0.000    ?          0.000    ?          ?        0.670     0.327     three
                 0.000    0.000    ?          0.000    ?          ?        0.668     0.102     zero
                 0.222    0.009    0.750      0.222    0.343      0.374    0.749     0.414     one
                 0.538    0.155    0.404      0.538    0.462      0.343    0.737     0.346     two
Weighted Avg.    0.603    0.292    ?          0.603    ?          ?        0.763     0.571

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 117   0   0   0   4 |   a = four
  26   0   0   1  14 |   b = three
   7   0   0   0   4 |   c = zero
  12   0   0   6   9 |   d = one
  17   0   0   1  21 |   e = two
```

PrincipalComponents with Bagging

```
=== Summary ===

Correctly Classified Instances         136               56.9038 %
Incorrectly Classified Instances       103               43.0962 %
Kappa statistic                          0.2714
Mean absolute error                      0.2239
Root mean squared error                  0.3452
Relative absolute error                 83.1748 %
Root relative squared error             94.0995 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.508    0.655      0.942    0.773      0.487    0.790     0.784     four
                 0.146    0.051    0.375      0.146    0.211      0.145    0.612     0.251     three
                 0.000    0.000    ?          0.000    ?          ?        0.682     0.083     zero
                 0.259    0.080    0.292      0.259    0.275      0.189    0.734     0.287     one
                 0.231    0.080    0.360      0.231    0.281      0.182    0.669     0.316     two
Weighted Avg.    0.569    0.288    ?          0.569    ?          ?        0.728     0.528

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 114   3   0   1   3 |   a = four
  24   6   0   4   7 |   b = three
   4   1   0   4   2 |   c = zero
  12   4   0   7   4 |   d = one
  20   2   0   8   9 |   e = two
```

PrincipalComponents with LMT

```
=== Summary ===

Correctly Classified Instances         140               58.5774 %
Incorrectly Classified Instances        99               41.4226 %
Kappa statistic                          0.2622
Mean absolute error                      0.2273
Root mean squared error                  0.3377
Relative absolute error                 84.4272 %
Root relative squared error             92.0605 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.983    0.627    0.617      0.983    0.758      0.452    0.815     0.801     four
                 0.000    0.000    ?          0.000    ?          ?        0.680     0.297     three
                 0.000    0.000    ?          0.000    ?          ?        0.646     0.082     zero
                 0.148    0.019    0.500      0.148    0.229      0.228    0.754     0.417     one
                 0.436    0.105    0.447      0.436    0.442      0.334    0.712     0.316     two
Weighted Avg.    0.586    0.337    ?          0.586    ?          ?        0.760     0.559

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 119   0   0   0   2 |   a = four
  32   0   0   2   7 |   b = three
   7   0   0   1   3 |   c = zero
  14   0   0   4   9 |   d = one
  21   0   0   1  17 |   e = two
```

AllRetained with Logistic

```
=== Summary ===

Correctly Classified Instances         181               75.7322 %
Incorrectly Classified Instances        58               24.2678 %
Kappa statistic                          0.6254
Mean absolute error                      0.1644
Root mean squared error                  0.2802
Relative absolute error                 61.0763 %
Root relative squared error             76.3767 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.169    0.852      0.950    0.898      0.788    0.950     0.933     four
                 0.463    0.056    0.633      0.463    0.535      0.464    0.854     0.585     three
                 0.000    0.000    ?          0.000    ?          ?        0.815     0.477     zero
                 0.630    0.052    0.607      0.630    0.618      0.569    0.881     0.576     one
                 0.769    0.080    0.652      0.769    0.706      0.646    0.911     0.672     two
Weighted Avg.    0.757    0.114    ?          0.757    ?          ?        0.913     0.769

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 115   6   0   0   0 |   a = four
  12  19   0   0  10 |   b = three
   3   0   0   7   1 |   c = zero
   4   1   0  17   5 |   d = one
   1   4   0   4  30 |   e = two
```

AllRetained with MultilayerPerceptron

```
=== Summary ===

Correctly Classified Instances         170               71.1297 %
Incorrectly Classified Instances        69               28.8703 %
Kappa statistic                          0.5599
Mean absolute error                      0.1285
Root mean squared error                  0.2882
Relative absolute error                 47.7431 %
Root relative squared error             78.5739 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.934    0.153    0.863      0.934    0.897      0.785    0.938     0.905     four
                 0.463    0.086    0.528      0.463    0.494      0.398    0.852     0.531     three
                 0.364    0.013    0.571      0.364    0.444      0.436    0.802     0.367     zero
                 0.481    0.052    0.542      0.481    0.510      0.452    0.897     0.617     one
                 0.538    0.100    0.512      0.538    0.525      0.430    0.862     0.524     two
Weighted Avg.    0.711    0.115    0.698      0.711    0.702      0.607    0.900     0.721

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 113   6   0   0   2 |   a = four
  11  19   0   0  11 |   b = three
   3   0   4   4   0 |   c = zero
   3   1   3  13   7 |   d = one
   1  10   0   7  21 |   e = two
```

AllRetained with Bagging

```
=== Summary ===

Correctly Classified Instances         168               70.2929 %
Incorrectly Classified Instances        71               29.7071 %
Kappa statistic                          0.5495
Mean absolute error                      0.1512
Root mean squared error                  0.2794
Relative absolute error                 56.1842 %
Root relative squared error             76.1694 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.934    0.127    0.883      0.934    0.908      0.809    0.947     0.902     four
                 0.512    0.091    0.538      0.512    0.525      0.430    0.841     0.529     three
                 0.091    0.018    0.200      0.091    0.125      0.107    0.770     0.224     zero
                 0.407    0.066    0.440      0.407    0.423      0.353    0.779     0.462     one
                 0.564    0.100    0.524      0.564    0.543      0.451    0.885     0.512     two
Weighted Avg.    0.703    0.105    0.684      0.703    0.692      0.602    0.892     0.693

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 113   7   0   0   1 |   a = four
   9  21   0   1  10 |   b = three
   2   1   1   6   1 |   c = zero
   3   1   4  11   8 |   d = one
   1   9   0   7  22 |   e = two
```

AllRetained with LMT

```
=== Summary ===

Correctly Classified Instances         179               74.8954 %
Incorrectly Classified Instances        60               25.1046 %
Kappa statistic                          0.6161
Mean absolute error                      0.145
Root mean squared error                  0.2686
Relative absolute error                 53.8522 %
Root relative squared error             73.2284 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.144    0.870      0.942    0.905      0.802    0.949     0.929     four
                 0.512    0.081    0.568      0.512    0.538      0.450    0.864     0.527     three
                 0.182    0.000    1.000      0.182    0.308      0.418    0.743     0.425     zero
                 0.593    0.047    0.615      0.593    0.604      0.555    0.849     0.615     one
                 0.667    0.085    0.605      0.667    0.634      0.560    0.906     0.647     two
Weighted Avg.    0.749    0.106    0.752      0.749    0.736      0.656    0.907     0.755

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 114   6   0   0   1 |   a = four
  10  21   0   0  10 |   b = three
   2   1   2   5   1 |   c = zero
   4   2   0  16   5 |   d = one
   1   7   0   5  26 |   e = two
```

Part 7 - Discussion

Overall, we were able to successfully train and test models on the *Students Performance Dataset*. Our Logistic model trained on the AllRetained subset produced the highest testing accuracy (75.7322%) of any of the 20 model/subset configurations we tested. As for other metrics permitting to this configuration, we achieved a Precision of 0.869, Recall of 0.757, F1 Score of 0.809, and area under the ROC curve of 0.913.

In our Logistic model trained on AllRetained, some of the data was not provided directly through Weka, but can be derived from other metrics. Here are our calculations for these cases:

Precision ((TP)/(TP+FP)): 0.869
TP = 0.757 (given)
FP = 0.114 (given)
Precision = (0.757)/(0.757+0.114) = 0.869

*F1-Score ((2*Precision*Recall)/(Precision+Recall)):* 0.809
Precision: 0.869 (derived)
Recall: 0.757 (given)
F1-Score = ((2*0.869*0.757)/(0.869+0.757)) = 0.809

Seeing that the accuracy of 75.7322% using a Logistic model trained on the AllRetained subset was the highest is indicative of the vast room for improvement in model performance.

Specifically, while this iteration of development solely examined accuracy ((TP+TN)/(TP+TN+FP+FN)), further performance analysis could consist of analyzing other metrics such as Precision ((TP)/(TP+FP)), Recall ((TP)/(TP+FN)), or the F1-Score ((2*Precision*Recall)/(Precision+Recall)).

Additionally, we believe that the AllRetained subset (in which no attributes were removed) produced the greatest accuracy largely due to the small volume of features in the dataset (12 after removing *GPA* and *StudentID* columns). Since the number of attributes was not particularly large, the Logistic model must have been able to learn nuanced relationships between each of the features—information otherwise lost after performing attribute selection algorithms upon the other four subsets (*CfsSubsetEval, CorrelationAttributeEval, PrincipalComponents,* and *ReliefFAttributeEval*).

Part 8 - Conclusion

As stated previously, we achieved the highest accuracy without removing attributes and by applying the Logistic classifier. We were able to construct a model that successfully predicts high-school students' GPA with a relatively high accuracy by evaluating features like parental education, absences, age, gender, ethnicity, study time, extracurriculars, and more. This analysis is an important first step in understanding various factors that influence high school students' academic performance. Furthermore, this information can help parents, educators, and policymakers make data-driven decisions that support student achievement.

In the future, the accuracy of this model can be improved upon by incorporating additional features like socioeconomic status, school environment, sleep patterns, and more. These factors can provide a more holistic view of a student's environment and habits, which may directly impact their academic performance. Additionally, we could also explore more advanced models like neural networks, which might capture interactions between features more effectively than a simple Logistic model. Tuning hyperparameters and experimenting with feature engineering could further optimize the model's performance.

Steps to Reproduce our Model:

1. In the Google Drive folder "Q1 Project - Dhruv and Soham," navigate to Datasets > AllRetained. Download the files "AllRetained_train.csv" and "AllRetained_test.csv."
2. Load "AllRetained_train.csv" onto Weka Explorer by clicking "Open file…" under the Preprocess tab.
3. Click "Save…" and save this dataset as "AllRetained_train.arff."
4. Repeat steps 2 and 3 by loading "AllRetained_test.csv" and saving it as "AllRetained_test.arff."
5. Now, open the file "AllRetained_train.arff" by clicking "Open file…" under the Preprocess tab.
6. Navigate to the "Classify" tab and click "Choose." Then, select the Logistic classifier under classifiers > functions > Logistic.
7. Under "Test options," select "Supplied test set" and click "Set…"

8. Click “Open file…” in the new pop-up window and select “AllRetained_test.arff.” Then, click “Close.”
9. Click the “Start” button. Weka will display the performance metrics for this model, which achieved an accuracy of 75.7322%.

Part 9 - Team Member Contributions

Dhruv: I worked on much of the technical aspects of this project. I constructed the algorithm for splitting the dataset into train, testing, and validation sets by using SciKit Learn’s *train_test_split* function. Additionally, I explored WEKA and it’s uses in employing both attribute selection algorithms and classifier models to train and test the various subsets of the data. One bottleneck I initially encountered was that many of the classifier models weren’t available to me for training/testing. The models that were available seemed to be more geared towards quantitative continuous data, such as a LinearRegression. This led me to believe that this was an issue related to the data type of the *GradeClass* class variable. Sure enough, *GradeClass* was of type *float* and had to be alphabetized into *strings*. So, I mapped each value (*0, 1, 2, 3, 4*) in *GradeClass* into its corresponding alphabetic form (*zero, one, two, three, four*). This new dataset with a stringified class column now allowed me to access many more classifiers correlated to nominal qualitative class variables.

Soham: In addition to writing the project overview and dataset information for this report, I also worked on preprocessing and displaying the results of our project. First, this involved downloading our dataset as a CSV from Kaggle and transferring it over to Weka to remove the ‘StudentID’ and ‘GPA’ attributes. Then, I saved this file as a CSV again and uploaded it onto Google Colab. Through creating a min_max_normalize method, I ensured that all of our attribute values were floats between 0 and 1. Next, using our results from CorrelationAttributeEval, ReliefFAttributeEval, CfsSubsetEval, and the PrincipalComponents attribute selection algorithms, I created four separate CSV files with the attributes that we retained. Also, I created a fifth CSV for the dataset of our choice, in which we did not remove any attributes. Following train-validation-test split and running model classifiers, I determined our best model by evaluating the accuracies. In our conclusion, I suggested potential directions for future work, such as incorporating additional attributes and exploring more complex ML models. I also wrote the steps to reproduce our best model on Weka using the train and test datasets for AllRetained.

Overall Takeaways: In this project, both of us learned how to effectively preprocess data, select relevant features through attribute selection, create multiple model classifiers on Weka, and evaluate our results in terms of performance metrics like accuracy, precision, recall, and area under the ROC curve. Additionally, we were able to apply what we learned in class to each part of the project. For example, during preprocessing, we applied our understanding of data cleaning techniques by removing non-essential and derived attributes like ‘StudentID’ and ‘GPA.’ We also implemented normalization techniques that we learned in class like min-max normalization to scale our attribute values between 0 and 1.

During attribution selection, we applied our knowledge from Lab 3 to identify which features to remove for each dataset. We also learned that certain algorithms are more effective at selecting the features that best predict class values. For example, our cutoff value was low using

CorrelationAttributeEval, suggesting that even the attributes we kept did not have a high correlation with the class values. However, during CfsSubsetEval, we observed that it not only prioritized features that had a higher predictive capability but also considered feature interactions.

References

Dataset:
https://www.kaggle.com/datasets/rabieelkharoua/students-performance-dataset

Scikit-learn train_test_split:
https://scikit-learn.org/1.5/modules/generated/sklearn.model_selection.train_test_split.html

Weka Documentation:
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CorrelationAttributeEval.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/ReliefFAttributeEval.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CfsSubsetEval.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/PrincipalComponents.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/Ranker.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/GreedyStepwise.html