Maltepe University
Faculty of Engineering and Natural Sciences
Software Engineering Department

CEN 420 01 Introduction to Pattern Recognition

Progress Report

Meaad Farag Bayuosef

***Abstract*—The "Student Performance Predictor" is a project undertaken by students from the Software Engineering Department at Maltepe University. It uses machine learning to predict student outcomes based on various personal and academic factors. The project dataset comprises 395 student records, and it employs Logistic Regression model,to ascertain the likelihood of students passing or failing their courses. This tool aims to enable educational institutions to intervene effectively to assist at-risk students, thereby enhancing educational outcomes.**

I. Introduction

The Student Performance Predictor project, harnesses advanced machine learning techniques to accurately forecast academic success and potential student failures. This sophisticated predictive tool is designed to serve as a critical asset for educational planners and administrators who are committed to improving student retention rates and academic performance. By leveraging detailed student data encompassing demographics, academic histories, and behavioral patterns, the project aims to identify students at risk of under performing early in their academic journey. This early identification allows for timely and targeted interventions, which are crucial in providing the necessary support to enhance students' educational outcomes. Moreover, by integrating empirical data analysis with predictive modeling, the initiative not only contributes to individual student success but also aids in the formulation of broader educational strategies that can be tailored to the needs of diverse student populations. Through this project, the team endeavors to bridge the gap between data science and educational insights, thereby fostering an environment where data-driven decision-making leads to tangible improvements in academic institutions.

II. Project Objectives

This project focuses on:

- Developing a predictive model that determines the likelihood of students passing or failing based on their academic and demographic data.
- Identifying key determinants of academic success to guide educational interventions.

III. Timeline and Milestones

- Weeks 1-3: Project conceptualization and data collection.
- Weeks 4-5: Data cleaning and initial preprocessing.
- Weeks 6-7: Feature engineering and analysis.
- Weeks 8-9: Model development and initial training.
- Weeks 10-11: Ongoing model evaluation and optimization (in progress).
- Weeks 12-15: Final documentation and project presentation (upcoming).

IV. Progress to Date

A. Data Collection

The dataset used in the "Student Performance Predictor" project comprises records for 395 students. It includes various features critical to understanding student performance, such as demographics, parental education levels, study habits, and historical academic performance. The data were sourced from the UCI Machine Learning Repository and have undergone rigorous preprocessing to ensure accuracy and relevance for model training. This preparation involved handling missing values, normalizing numerical inputs, and encoding categorical variables to create a robust dataset for our machine learning algorithms.

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	passed
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	no	no	4	3	4	1	1	3	6	no
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	yes	no	5	3	3	1	1	3	4	no
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	yes	no	4	3	2	2	3	3	10	yes
3	GP	F	15	U	GT3	T	4	2	health	services	...	yes	yes	3	2	2	1	1	5	2	yes
4	GP	F	16	U	GT3	T	3	3	other	other	...	no	no	4	3	2	1	2	5	4	yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	services	services	...	no	no	5	5	4	4	5	4	11	no
391	MS	M	17	U	LE3	T	3	1	services	services	...	yes	no	2	4	5	3	4	2	3	yes
392	MS	M	21	R	GT3	T	1	1	other	other	...	no	no	5	5	3	3	3	3	3	no
393	MS	M	18	R	LE3	T	3	2	services	other	...	yes	no	4	4	1	3	4	5	0	yes
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	yes	no	3	2	3	3	3	5	5	no

395 rows × 31 columns

Fig. 1. Dataset

B. Data Preprocessing

- Handling Missing Values: Missing data were imputed using appropriate statistical methods to maintain data integrity.
- Data Encoding: Categorical variables were transformed into numerical formats through label encoding to facilitate their use in our models.
- Feature Scaling and Normalization: Numerical features underwent standardization (subtracting the mean and scaling to unit variance) and Min-Max scaling to ensure uniformity in range and scale, enhancing model performance.

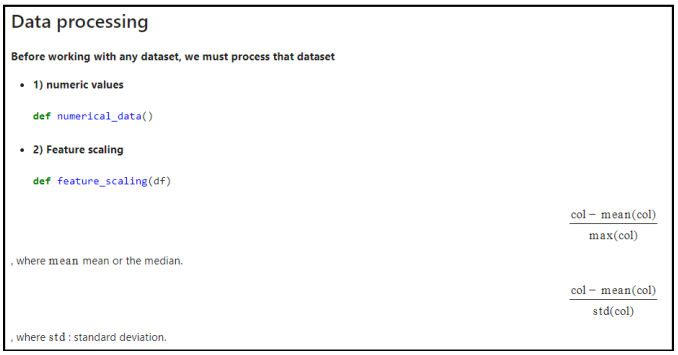

Fig. 2. Data Preprocessing

C. Feature Engineering:

We refined the dataset for machine learning through detailed feature engineering:

- Exploratory Data Analysis (EDA): Conducted to uncover key predictors by analyzing data distributions and correlations.
- Data Transformations:
  - Encoding: Transformed categorical variables like 'school' and 'sex' into numerical formats using label encoding to facilitate algorithm processing.
  - Normalization and Scaling: Applied z-score normalization to standardize features like 'age', and used Min-Max scaling for bounded features, ensuring all variables contribute equally to model performance.

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	passed
0	0	1	18	0	1	1	4	4	3	0	...	0	0	4	3	4	1	1	3	6	0
1	0	1	17	0	1	0	1	1	3	4	...	1	0	5	3	3	1	1	3	4	0
2	0	1	15	0	0	0	1	1	3	4	...	1	0	4	3	2	2	3	3	10	1
3	0	1	15	0	1	0	4	2	1	2	...	1	1	3	2	2	1	1	5	2	1
4	0	1	16	0	1	0	3	3	4	4	...	0	0	4	3	2	1	2	5	4	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	1	0	20	0	0	1	2	2	2	2	...	0	0	5	5	4	4	5	4	11	0
391	1	0	17	0	0	0	3	1	2	2	...	1	0	2	4	5	3	4	2	3	1
392	1	0	21	1	1	0	1	1	4	4	...	0	0	5	5	3	3	3	3	3	0
393	1	0	18	1	0	0	3	2	2	4	...	1	0	4	4	1	3	4	5	0	1
394	1	0	19	0	0	0	1	1	4	3	...	1	0	3	2	3	3	3	5	5	0

395 rows × 31 columns

Fig. 3. Numerical data

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	passed
0	0.0	1.0	0.059264	0.0	1.0	1.0	1.00	1.00	0.75	0.00	...	0.0	0.0	0.75	0.50	0.75	0.00	0.00	0.50	0.003882	0.0
1	0.0	1.0	0.013809	0.0	1.0	0.0	0.25	0.25	0.75	1.00	...	1.0	0.0	1.00	0.50	0.50	0.00	0.00	0.50	-0.022785	0.0
2	0.0	1.0	-0.077100	0.0	0.0	0.0	0.25	0.25	0.75	1.00	...	1.0	0.0	0.75	0.50	0.25	0.25	0.50	0.50	0.057215	1.0
3	0.0	1.0	-0.077100	0.0	1.0	0.0	1.00	0.50	0.25	0.50	...	1.0	1.0	0.50	0.25	0.25	0.00	0.00	1.00	-0.049451	1.0
4	0.0	1.0	-0.031646	0.0	1.0	0.0	0.75	0.75	1.00	1.00	...	0.0	0.0	0.75	0.50	0.25	0.00	0.25	1.00	-0.022785	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	1.0	0.0	0.150173	0.0	0.0	1.0	0.50	0.50	0.50	0.50	...	0.0	0.0	1.00	1.00	0.75	0.75	1.00	0.75	0.070549	0.0
391	1.0	0.0	0.013809	0.0	0.0	0.0	0.75	0.25	0.50	0.50	...	1.0	0.0	0.25	0.75	1.00	0.50	0.75	0.25	-0.036118	1.0
392	1.0	0.0	0.195627	1.0	1.0	0.0	0.25	0.25	1.00	1.00	...	0.0	0.0	1.00	1.00	0.50	0.50	0.50	0.50	-0.036118	0.0
393	1.0	0.0	0.059264	1.0	0.0	0.0	0.75	0.50	0.50	1.00	...	1.0	0.0	0.75	0.75	0.00	0.50	0.75	1.00	-0.076118	1.0
394	1.0	0.0	0.104718	0.0	0.0	0.0	0.25	0.25	1.00	0.75	...	1.0	0.0	0.50	0.25	0.50	0.50	0.50	1.00	-0.009451	0.0

395 rows × 31 columns

Fig. 4. Features scalling

D. Data visualization

Data visualization plays a crucial role in our project by allowing us to explore the data graphically, spot trends, understand the distribution, and make informed decisions about feature selection and model tuning. Throughout the project, we utilized various visualization techniques to aid in both data preprocessing and analysis phases.

features visualisation:

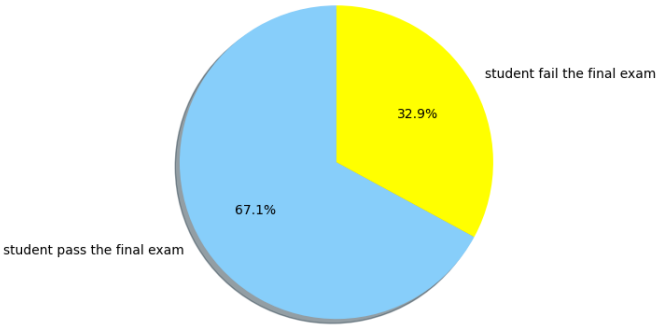

Fig. 5. students pass/fail

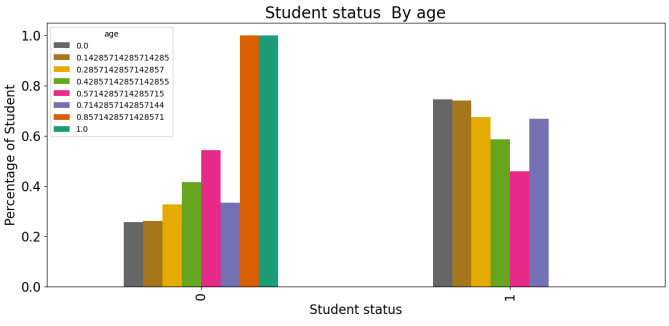

Fig. 7. Student status by age

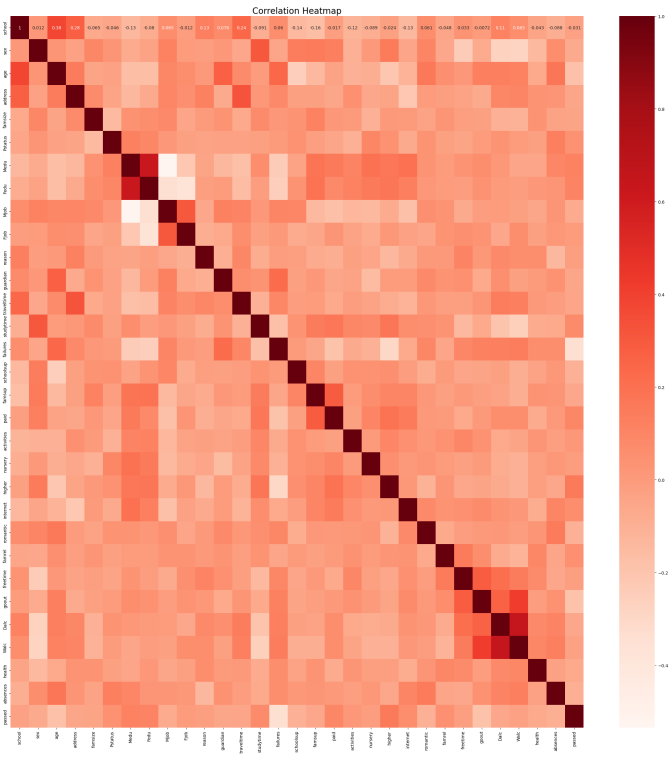

Fig. 6. Correlation heatmap

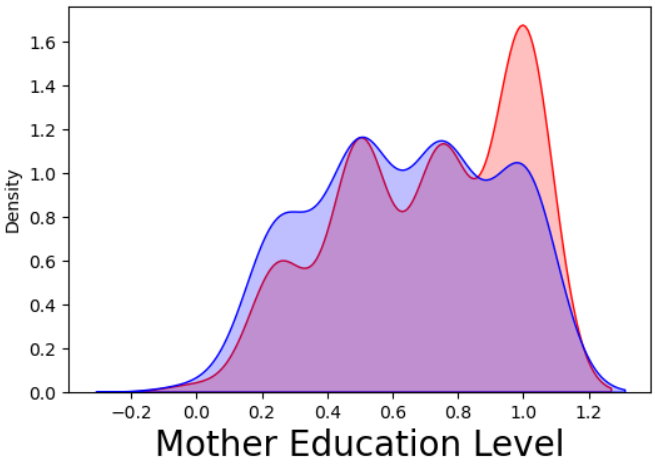

Fig. 8. Mother education

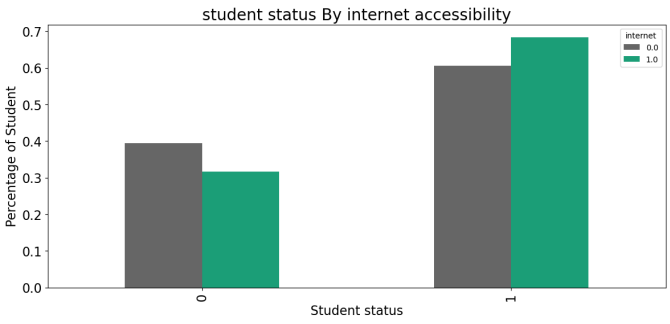

Fig. 9. internet accessibility

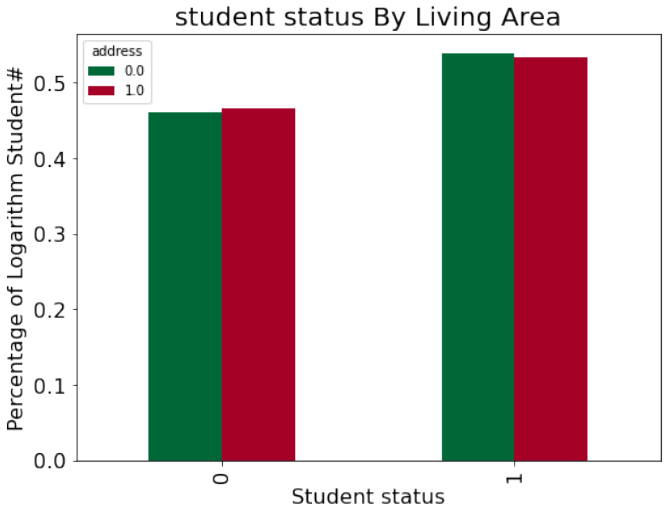

Fig. 10. student status By Living Area

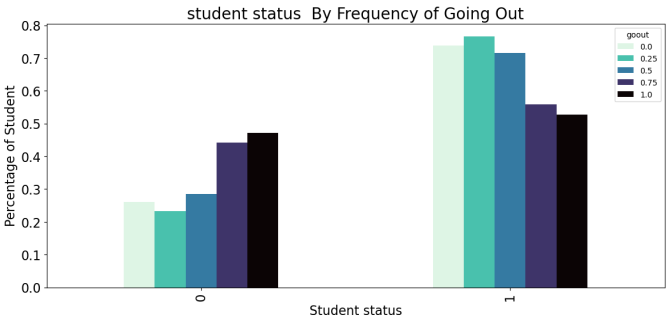

Fig. 11. going out

1) General conclusion: After dealing with the most relevant features, a high-achieving student with excellent conditions for academic potential is likely to have this profile :

- Not going out with friends frequently
- Not in a romantic relationship
- Parents receive higher education,especially mother
- Having a strong desire to pursue higher education
- The mother is a healthcare professional
- The father is a teacher
- Not missing classes
- Having access to the internet
- Studying more than 10 hours a week
- Healthy

E. Model Training Process

During training phase of the ”Student Performance Predictor” project, three distinct machine learning algorithms were utilized to determine their effectiveness in predicting student performance. The algorithms selected were Logistic Regression, Support Vector Machines (SVM), and Decision Trees. Each algorithm was chosen for its unique characteristics and suitability for binary classification tasks.

Implementation and Evaluation of Models:

1) **Logistic Regression:**

- *Implementation:* Logistic Regression was implemented using the `LogisticRegression` class from the `scikit-learn` library. This model is particularly suitable for binary classification tasks where the outcome is categorical.
- *Training:* The model was trained on a preprocessed dataset where categorical variables were encoded, and numerical features were scaled. The training process involved maximizing the log-likelihood of the training data, effectively fitting a logistic curve to predict the probability of a student passing.
- *Accuracy:* The Logistic Regression model achieved an accuracy of 0.638 on the test set.

2) **Support Vector Machine (SVM):**

- *Implementation:* SVM was deployed using the `SVC` class from `scikit-learn`, configured with a radial basis function (RBF) kernel to handle non-linear boundaries between classes.
- *Training:* SVM training involved finding the hyperplane that best separates the classes with the maximum margin. The model's parameters were fine-tuned using grid search to optimize performance.
- *Accuracy:* The SVM model recorded an accuracy of 0.613, indicating its performance with the given feature set and kernel configuration.

3) **Decision Tree:**

- *Implementation:*The decision tree model was implemented using `DecisionTreeClassifier`from `scikit-learn`. Decision Trees are known for their ability to form branching structures, making decisions based on feature thresholds.
- *Training:* The model was trained by recursively splitting the training data into subsets based on feature values that result in the highest information gain. The process continues until the leaves are pure or a stopping criterion is met.
- *Accuracy:* The Decision Tree model achieved an accuracy of 0.630.

```
Logistic Regression Accuracy: 0.6386554621848739
              precision    recall  f1-score   support

         0.0       0.71      0.24      0.36        50
         1.0       0.63      0.93      0.75        69

    accuracy                           0.64       119
   macro avg       0.67      0.58      0.55       119
weighted avg       0.66      0.64      0.58       119

SVM Accuracy: 0.6134453781512605
              precision    recall  f1-score   support

         0.0       1.00      0.08      0.15        50
         1.0       0.60      1.00      0.75        69

    accuracy                           0.61       119
   macro avg       0.80      0.54      0.45       119
weighted avg       0.77      0.61      0.50       119

Decision Tree Accuracy: 0.6302521008403361
              precision    recall  f1-score   support

         0.0       0.58      0.42      0.49        50
         1.0       0.65      0.78      0.71        69

    accuracy                           0.63       119
   macro avg       0.62      0.60      0.60       119
weighted avg       0.62      0.63      0.62       119
```

Fig. 12. evaluate each model

V. Results and Findings:

- Upon comparing the accuracies, Logistic Regression was chosen for further development and fine-tuning, given its higher accuracy and the model's interpretability, which is crucial for stakeholders to understand the factors influencing predictions.
- Preliminary testing shows significant factors influencing performance include parental education levels, student aspirations, and engagement in academic activities.

VI. Challenges and Solutions

- Model Selection: Selecting the optimal model was a key challenge due to the need to balance predictive accuracy with model interpretability. This challenge was addressed by implementing three different machine learning models—Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Preliminary evaluations were conducted based on accuracy, precision, recall, and F1-score directly from initial test results, without explicit cross-validation. This approach allowed for an immediate comparison of model performance to determine which model best suits the project requirements.
- Data Constraints: The dataset comprised only 395 student records, which posed challenges in terms of building highly robust models. To address this, the project employed techniques like feature scaling and data preprocessing to improve model training effectiveness under the constraint of limited data.

VII. Next Steps and Future Work

- Detailed Model Evaluation: Continue refining the chosen Logistic Regression model by tuning its parameters and validating its performance on unseen data.
- Final Reporting: Preparation of comprehensive project documentation, including detailed insights into the data analysis and model selection process and Prepare for the final presentation and submission

VIII. Conclusion

The "Student Performance Predictor" project has effectively developed and evaluated three machine learning models—Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—using a dataset of 395 student records. Logistic Regression was selected for further refinement due to its optimal balance of simplicity and predictive accuracy. The project's next steps include fine-tuning this model to enhance its generalization capabilities and preparing comprehensive documentation to detail the findings and methodologies used. These efforts aim to provide educational stakeholders with actionable insights to improve student academic outcomes.

IX. References

1) UCI Machine Learning Repository.Student Performance Data Set. https://archive.ics.uci.edu/ml/datasets/student+performance
2) Hosmer Jr, D. W., et al. (2013). *Applied Logistic Regression*. John Wiley & Sons. https://www.wiley.com
3) Altman, N. S. (1992). *The American Statistician*. https://www.tandfonline.com
4) Cortes, C., & Vapnik, V. (1995). *Machine Learning*, 20(3), 273-297. https://link.springer.com
5) James, G., et al. (2013). *An Introduction to Statistical Learning*. Springer.
6) Pedregosa, F., et al. (2011). *Journal of Machine Learning Research*, 12, 2825-2830. https://jmlr.csail.mit.edu
7) Google. https://www.google.com
8) AnalyticVue on neural networks and machine learning. https://www.analyticvue.com

X. Appendices

```
# mapping strings to numeric values:
def numerical_data():
    df['school'] = df['school'].map({'GP': 0, 'MS': 1})
    df['sex'] = df['sex'].map({'M': 0, 'F': 1})
    df['address'] = df['address'].map({'U': 0, 'R': 1})
    df['famsize'] = df['famsize'].map({'LE3': 0, 'GT3': 1})
    df['Pstatus'] = df['Pstatus'].map({'T': 0, 'A': 1})
    df['Mjob'] = df['Mjob'].map({'teacher': 0, 'health': 1, 'services': 2, 'at_home': 3, 'other': 4})
    df['Fjob'] = df['Fjob'].map({'teacher': 0, 'health': 1, 'services': 2, 'at_home': 3, 'other': 4})
    df['reason'] = df['reason'].map({'home': 0, 'reputation': 1, 'course': 2, 'other': 3})
    df['guardian'] = df['guardian'].map({'mother': 0, 'father': 1, 'other': 2})
    df['schoolsup'] = df['schoolsup'].map({'no': 0, 'yes': 1})
    df['famsup'] = df['famsup'].map({'no': 0, 'yes': 1})
    df['paid'] = df['paid'].map({'no': 0, 'yes': 1})
    df['activities'] = df['activities'].map({'no': 0, 'yes': 1})
    df['nursery'] = df['nursery'].map({'no': 0, 'yes': 1})
    df['higher'] = df['higher'].map({'no': 0, 'yes': 1})
    df['internet'] = df['internet'].map({'no': 0, 'yes': 1})
    df['romantic'] = df['romantic'].map({'no': 0, 'yes' : 1})
    df['passed'] = df['passed'].map({'no': 0, 'yes': 1})
    # reorder dataframe columns :
    col = df['passed']
    del df['passed']
    df['passed'] = col
```

Fig. 13. numerical data

```
# feature scaling will allow the algorithm to converge faster, large data will have same scal
def feature_scaling(df):
    for i in df:
        col = df[i]
        # let's choose columns that have large values
        if(np.max(col)>6):
            Max = max(col)
            Min = min(col)
            mean = np.mean(col)
            col  = (col-mean)/(Max)
            df[i] = col
        elif(np.max(col)<6):
            col = (col-np.min(col))
            col /= np.max(col)
            df[i] = col
```

Fig. 14. feature scaling

1) data inspection

```
[16]: df.shape

[16]: (395, 31)

[17]: df.dropna().shape # their is no null value "fortunately:)"

[17]: (395, 31)

[18]: df.columns

[18]: Index(['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu',
             'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
             'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
             'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc',
             'Walc', 'health', 'absences', 'passed'],
            dtype='object')

[19]: features=['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu',
              'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
              'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
              'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc',
              'Walc', 'health', 'absences']
```

Fig. 15. data inspection