**EC 9560 – DATA MINING**

**LAB 04**

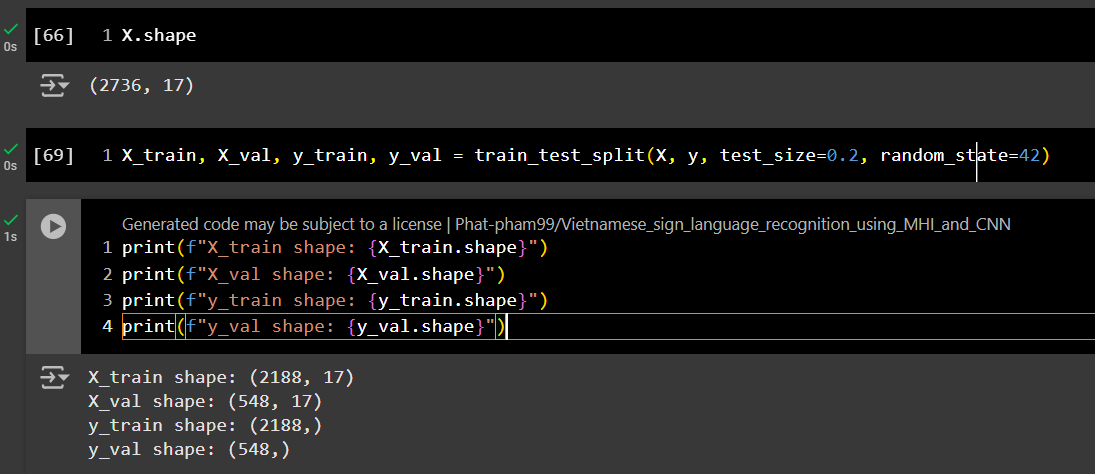
**DARMILA.T**

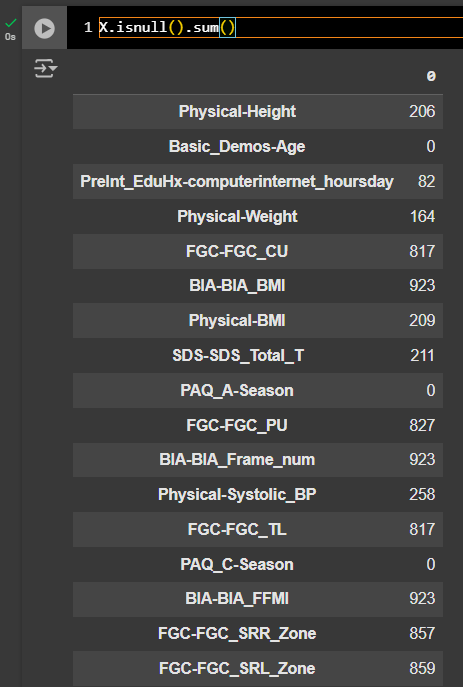
**2020/E/027**

**SEMESTER 07**

**18th DECEMBER 2024**

**Data Splitting**

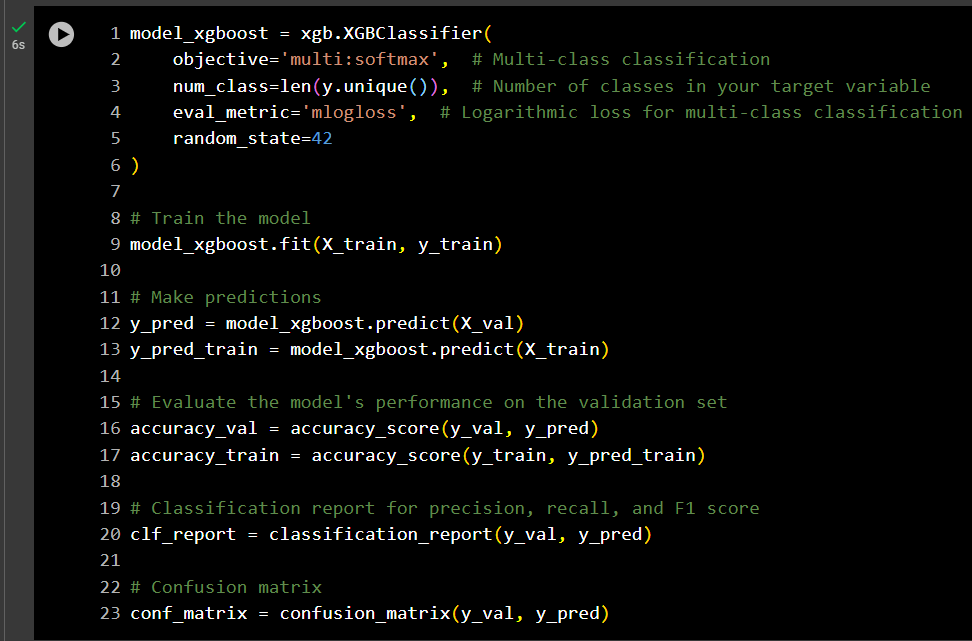
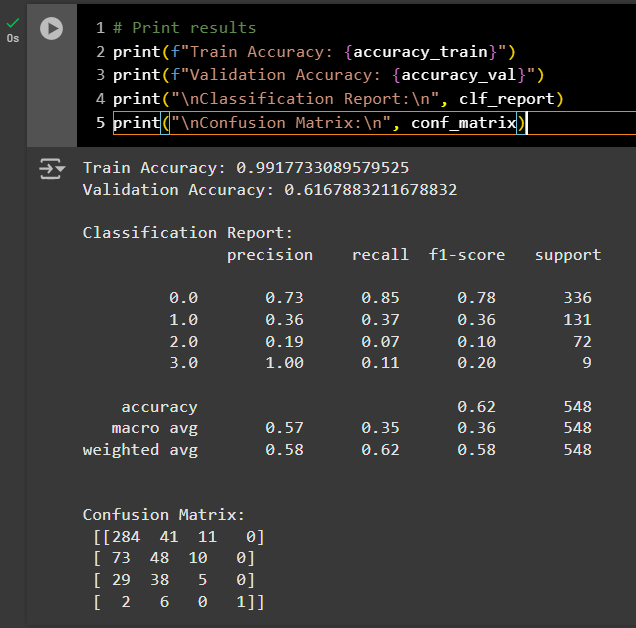
****

Split the train data into train and validation sets, and test data for final evaluation of the model selection.

There are still missing values in the feature selection of the training data, but we cannot remove them, as some features have a larger number of missing values. Therefore, I have decided to train the XGBoost model, which can handle missing data effectively.

**XGBoost model**

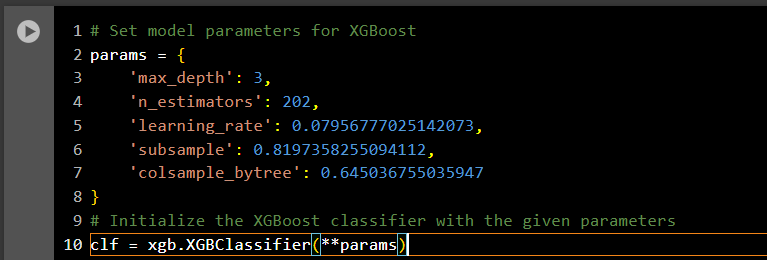
The XGBClassifier is used for multi-class classification. The objective='multi:softmax' parameter specifies that we're solving a multi-class classification problem, and num\_class specifies the number of unique classes in the target variable y.



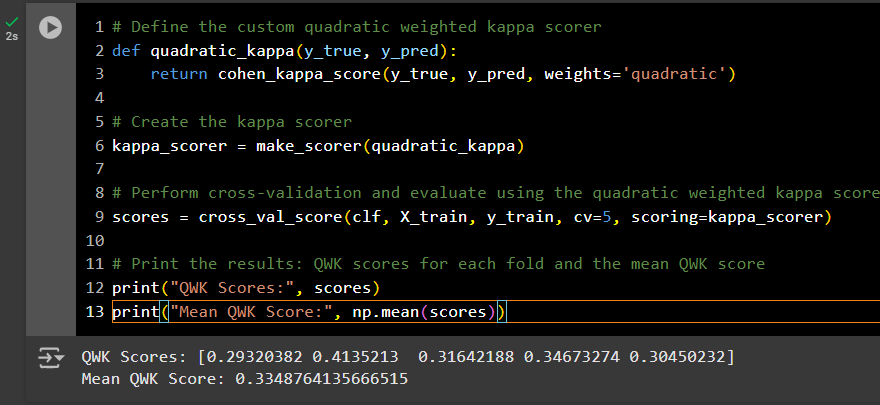
Here, the validation and training accuracy deviate significantly, indicating overfitting. To address this issue, I have decided to adjust the parameters of the XGBoost model.

I define the hyperparameters for the XGBClassifier. These parameters are passed when creating the model.

* max\_depth: Maximum depth of a tree, controlling overfitting.
* n\_estimators: Number of boosting rounds (trees).
* learning\_rate: Step size shrinkage to prevent overfitting.
* subsample: Fraction of samples used for fitting each tree.
* colsample\_bytree: Fraction of features to use when building each tree.

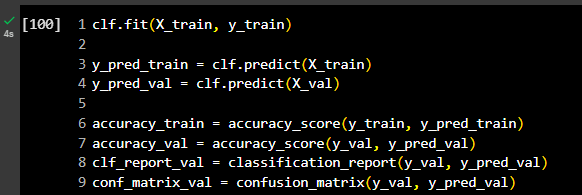


Prints the **quadratic weighted kappa** scores for each fold of cross-validation and their mean.



This function calculates the **quadratic weighted kappa** score, which is a metric used to evaluate the agreement between two raters (between the true and predicted values).

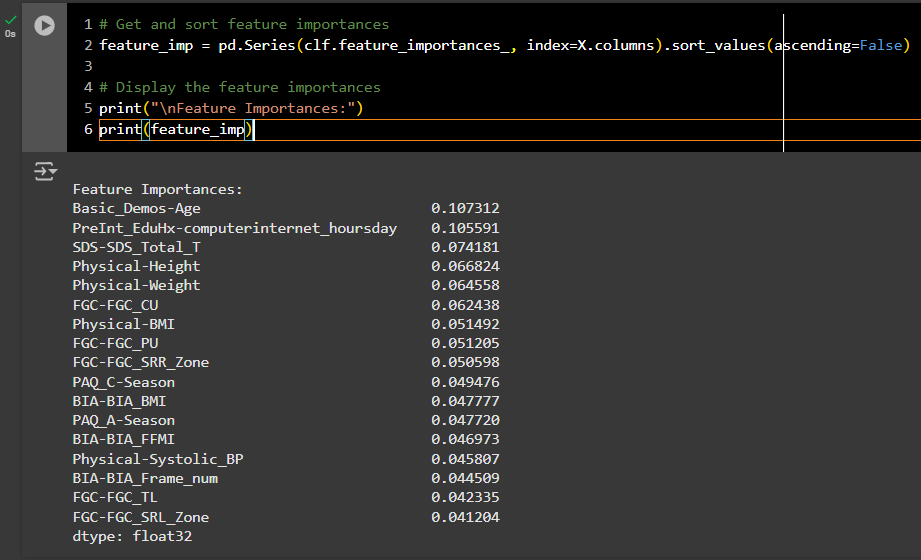
Ensures that each fold of the cross-validation maintains the same proportion of classes as the original dataset (for imbalanced class distributions).



A screenshot of a computer

Description automatically generated

Here, we can observe that the training accuracy has been reduced, and the gap between the training and validation accuracies is smaller. Additionally, this method allows us to see the importance of each feature. This approach is particularly useful for evaluating multi-class classification models and understanding the relative importance of each feature in making predictions.



Here, age and computer and internet usage are the most influential features in classifying the level of problematic internet use among children and adolescents.

**Handling the missing values**  
A screen shot of a black background

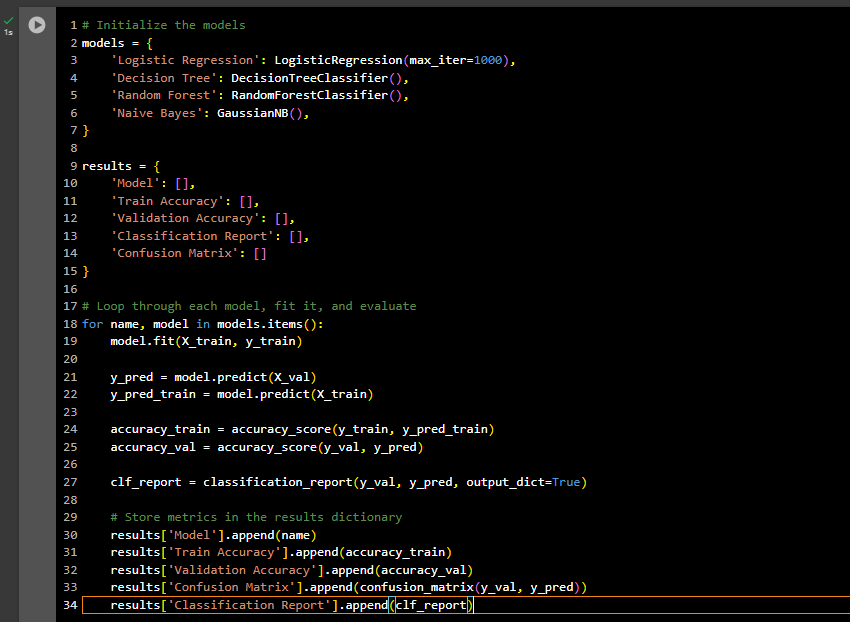
Description automatically generated

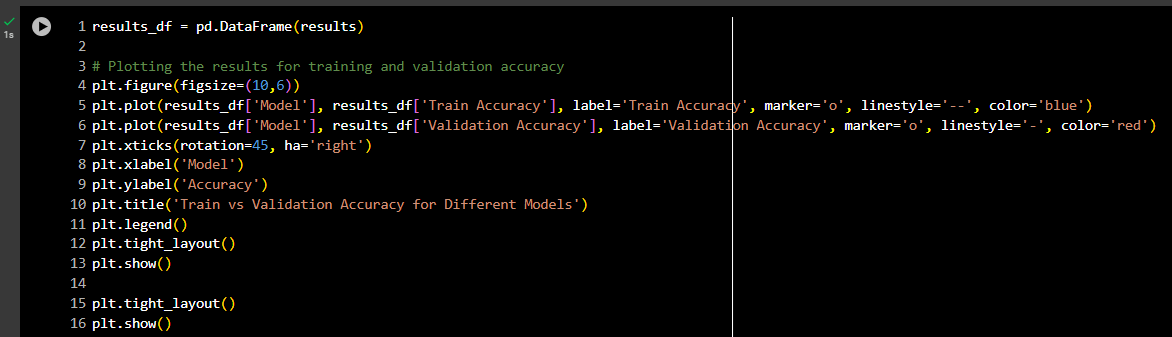
Here, I use KNN imputation for numerical features and mode imputation for categorical features.

**Train the model for different algorithms**

Here, I used the following algorithms to build the model:

* Logistic Regression with a maximum of 1000 iterations to ensure convergence.
* Decision Tree Classifier.
* Random Forest Classifier.
* Naive Bayes Classifier (GaussianNB).

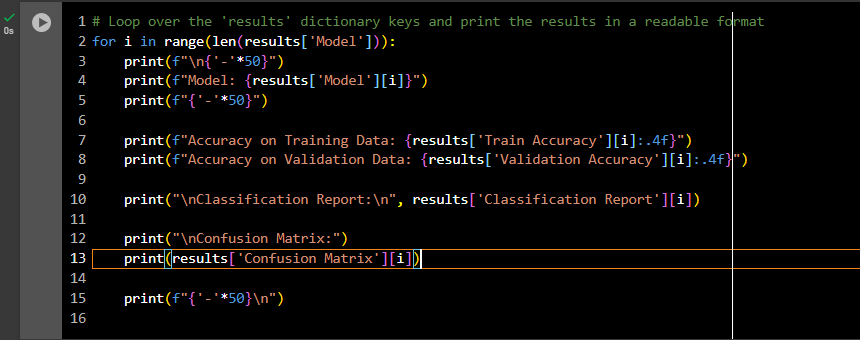




A graph with red lines and blue dots

Description automatically generated

Here, we can see the difference between each model. Logistic Regression and Naive Bayes provide the most balanced models, while Decision Tree and Random Forest models exhibit overfitting.

The evaluation matrix of models:

A rectangular black rectangle with white lines

Description automatically generated

A black rectangular object with white lines

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Model | Train Accuracy | Validation Accuracy |
| XGBoost | 0.779 | 0.618 |
| Logistic Regression | 0.6024 | 0.6259 |
| Decision Tree | 0.9991 | 0.4617 |
| Random Forest | 0.9991 | 0.5766 |
| Naïve Bayes | 0.5548 | 0.5383 |

**Analysis:**

* **XGBoost** shows a good balance between training accuracy (0.779) and validation accuracy (0.618). It indicates that the model has learned well from the training data and generalizes moderately to unseen data.
* **Logistic Regression** has lower training accuracy (0.6024) but performs better on the validation set (0.6259) compared to other models. This suggests that logistic regression is less prone to overfitting, and it generalizes better on unseen data.
* ***Decision Tree*** *and* ***Random Forest*** *both show extremely high training accuracy (0.9991), which suggests they are overfitting the training data. Their validation accuracies (0.4617 and 0.5766, respectively) are much lower, indicating poor generalization.*
* ***Naïve Bayes*** *has low training and validation accuracy (0.5548 and 0.5383, respectively), indicating that this model is not performing well for the given problem.*

In this case, **Logistic Regression** had missing values imputed, while **XGBoost** was trained without imputing missing values.

**Best Model: XGBoost**

* **Why XGBoost?**
  + **Robustness to Missing Data**: XGBoost's ability to handle missing values natively and still achieve relatively high performance (especially the validation accuracy of 0.618) makes it an excellent model in this case, especially given that the missing values were not imputed before training.
  + **High Validation Accuracy**: Although Logistic Regression showed a slightly higher validation accuracy (0.6259), the fact that XGBoost performed well without imputation is a strong point in its favor, as it demonstrates its robustness and flexibility in real-world scenarios where missing data might be common.
  + **Higher Training Accuracy**: XGBoost also has higher training accuracy compared to Logistic Regression, indicating that it has learned better patterns from the data.

**Conclusion**

While Logistic Regression performs well after imputing missing values, XGBoost stands out as the better choice due to its robustness in missing data, better handling of the feature interactions, and higher training accuracy. Given that XGBoost was able to achieve high validation accuracy without any imputation, it is likely to be more effective in scenarios with missing data, making it the best model overall in this case.