Assignment 2

Name: Aris Podotas

University: National and Kapodistrian University of Athens

Program: Data Science and Information Technologies

Specialization: Bioinformatics - Biomedical Data

Lesson: Machine Learning In Computational Biology

Date: May 2025

Contents

Abstract

The task of properly classifying tumors into benign and malignant categories using machine learning has been a focal point of the 21^{st} century. Here we use the advances of machine learning as a field to try and propose methods of completing this classification in a non trivial way.

1 Introduction

In classification task like the one of categorizing tumors into benign and malignant categories, the trivial solution is to focus on the inequality of the two sets. Should a model always predict a benign tumor it would have a good accuracy over all instances due to the how frequent the benign tumors are over the malignant ones. For this reason this classification task remains prominent in the machine learning space. It is critical to develop reliable methods for doing this classification since domain knowledge takes years to attain and mistakes are critical and costly. Benign tumors can appear in all tissues through ones lifetime, at any points a multitude of factors may convert a benign tumor to a malignant one [1], [2].

Catching malignant tumors early can be crucial for the survival and cost of treatment for women with breast cancer and thus it is important to have apt methodology that can accurately distinguish between benign and malignant tumors [3]. Machine learning models have been developed for calcifications of breast cancer tumors ranging from mammography, ultrasound, MRI, histology, and thermography [4]. Images of fine needle aspirates as in our data will contribute to the available methods of classification for this cause.

2 Methods

2.1 Implementation

All results were created in the *Python* programming language [5]. Functionality from this repository (last assignment) was also used in the analysis.

All source files will be available in this repository.

The approach of the implementation was of OOP (Object Oriented Programming) and all requirements are completed within two classes, a utility class (implemented here) and a RNCV (Repeated Nested Cross Validation) class (implemented here).

2.2 Models

Models:

1. Logistic Regression
2. Gaussian Naive Bayes
3. Linear Discriminant Analysis
4. Support Vector Classifier
5. Random Forest Classifier

Models were imported from the *sklearn* [6] library in *Python* [5].

2.3 Evaluation metrics

The following evaluation metrics were chosen:

1. balanced accuracy score
2. f1 score
3. Hamming loss

4. fbeta score
5. Jaccard score
6. Matthews corrcoef
7. precision score
8. recall score

We will not be using the "Auc" metrics since our task is not fit for it. For the fbeta score we have used a beta of 2 since we would like to give the negative and less represented class more weight.

2.4 Confidences

Repeated nested K-fold cross validation for 10 folds, 5 outer folds and 3 inner folds were done for each model as per the requirements. Each individual evaluation of any model at any stage is calculated but only kept in memory momentarily, the results of each evaluation is seen only within the plots produced by the class. Individual evaluations from the inner folds outer folds and each loop are calculated and plotted in boxplots that try to capture the distribution space of each models evaluation rather than singular evaluation values.

2.5 Figures

Figures were generated from the *matplotlib* [7] library, all implementations are available at our source files.

In all box plots the dotted green line denotes the mean value and the orange line denotes the median value.

2.6 Missing data

Since missing data was found our strategy for the handling of such entries will be to replace the entry with the median value of the feature in questions. We will choose the median to be more resilient to outliers since we will see that the mean has been affected over the median.

2.7 Classes

A utility class (implemented here) and a RNCV (Repeated Nested Cross Validation) class (implemented here) were written according to the following diagram.

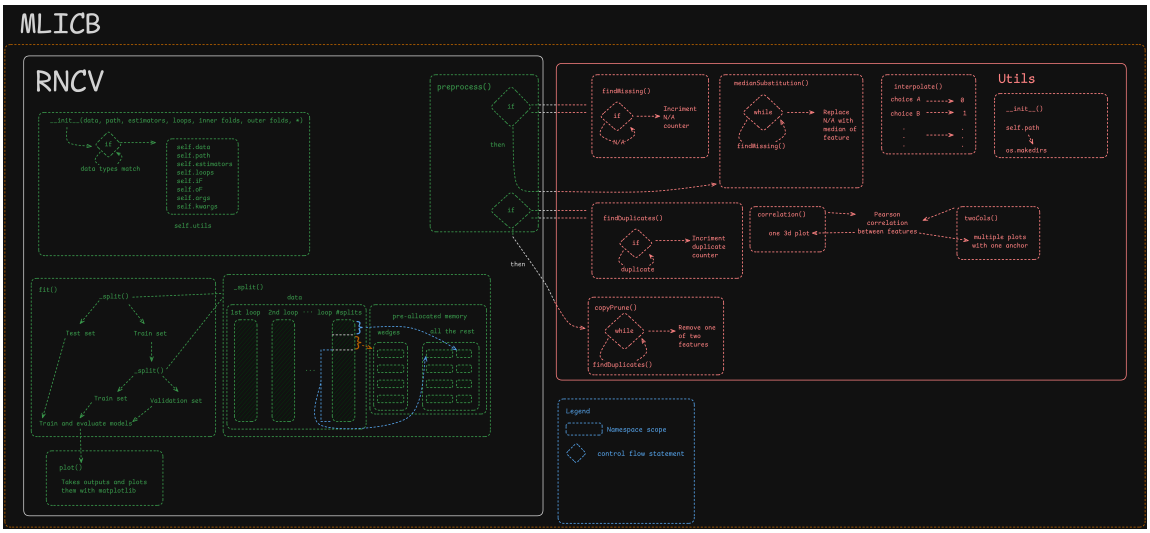

Figure 1: Flow chart of class implementation

This figure was made in Excalidraw. Only the main aspects of the class are written in the flow diagram since other implementation details are not part of the purpose of the class such as the "__repr__()" magic method.

The "RNCV" class has an attribute of a "Utility" class instance object and that is why these two have control flow from one into the other.

2.8 Data transformations

Most of our data is of continuous variables that need no interpolation of any sort considering there are fields for mean values standard errors and "worst cases". We will convert the diagnosis field to a binary values one where:

1. B $\rightarrow$ 0
2. M $\rightarrow$ 1

2.9 Feature selection

For the main requirements of the assignment a Principled Component Analysis was used (after the baseline models). We will require a 95% explain of our data variance after the feature selection.

For further feature selections see 3.4

2.10 Hyper parameter optimization

The optuna library [8] was used for all hyperparameter tuning (only in an instance after the baseline). All optimizations are in relation to the fbeta score since we would like to define the beta parameter of this metric to 2 for a higher weight on the smaller positive class.

3 Results

3.1 Data Exploration

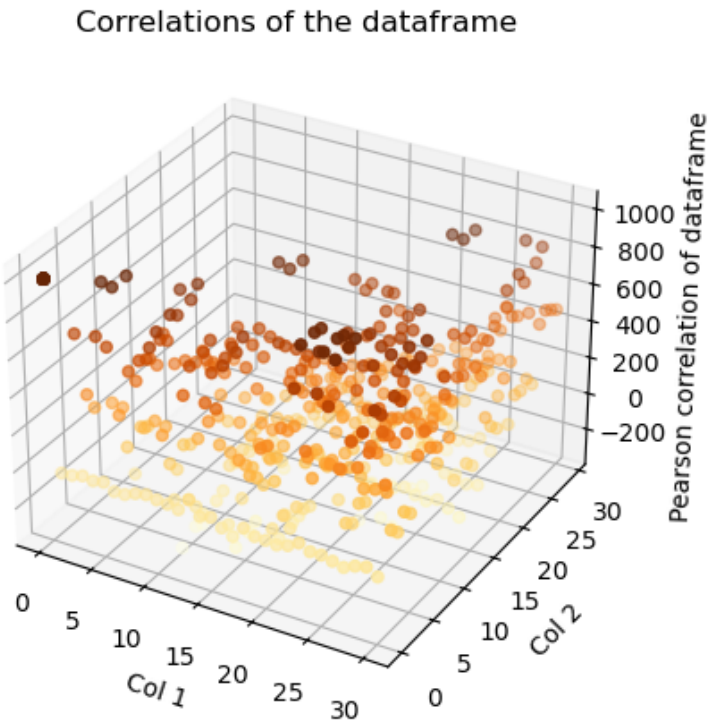

Figure 2: Pearson correlation of all feature pairs

We can see that there is a lot of pruning to be done for features that are correlated in Figure 2.

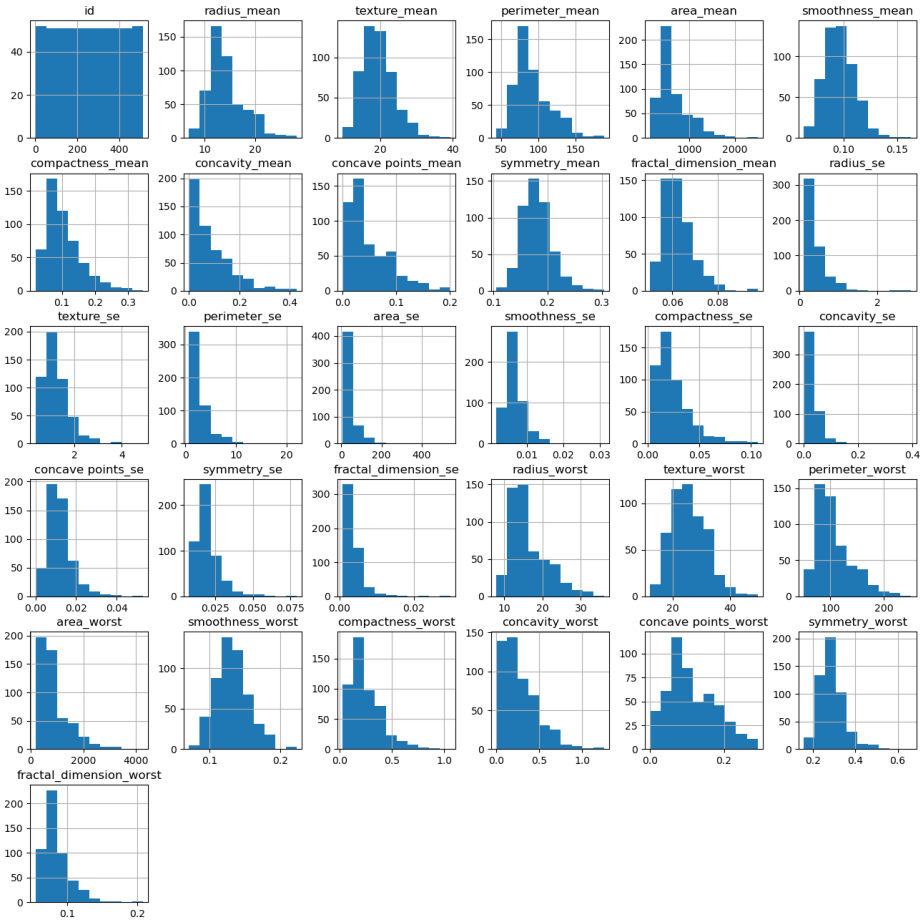

Figure 3: Distribution of values for each feature

We can make some sort of comment on the appearance of the central limit theorem in Figure 3 considering we get a Gaussian looking distribution [9] from taking the mean values of feature's distributions.

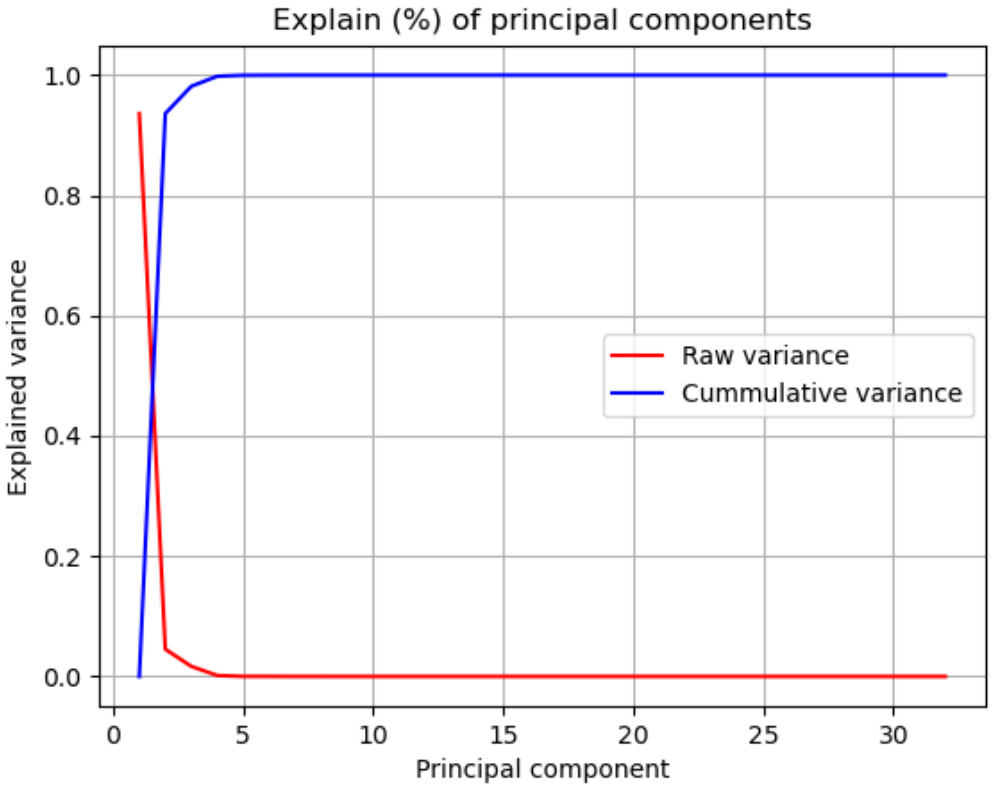

Figure 4: Explain % of all principal components

We can see that very few principled components will be needed for the 95% variance explain we will require. Specifically we will see that only 2 features reach this percentage (other than the labels).

Descriptive statistics were calculated for the initial data in the respective notebook and then saved to this comma separated values file.

3.2 Baseline

We are tasked with evaluating this data completely thus we need to identify data types and cast fields to values we can use. Our data was of continuous values variables for all fields but one, that was converted to a binary field as was mentioned in 2.8. Immediately after and with no feature selection or optimization the baseline model was made.

Evaluations of the Baseline:

Figure 5: Final distributions of all loops of the baseline model

You can find the baseline model in in this folder. As was said in 2.10 one of the best descriptive statistics here is the fbeta score. It is easy for our models to simply always predict benign and get high metrics but the fbeta score is weighed toward the proper predictions of the malignant tumors.

3.3 Feature selection

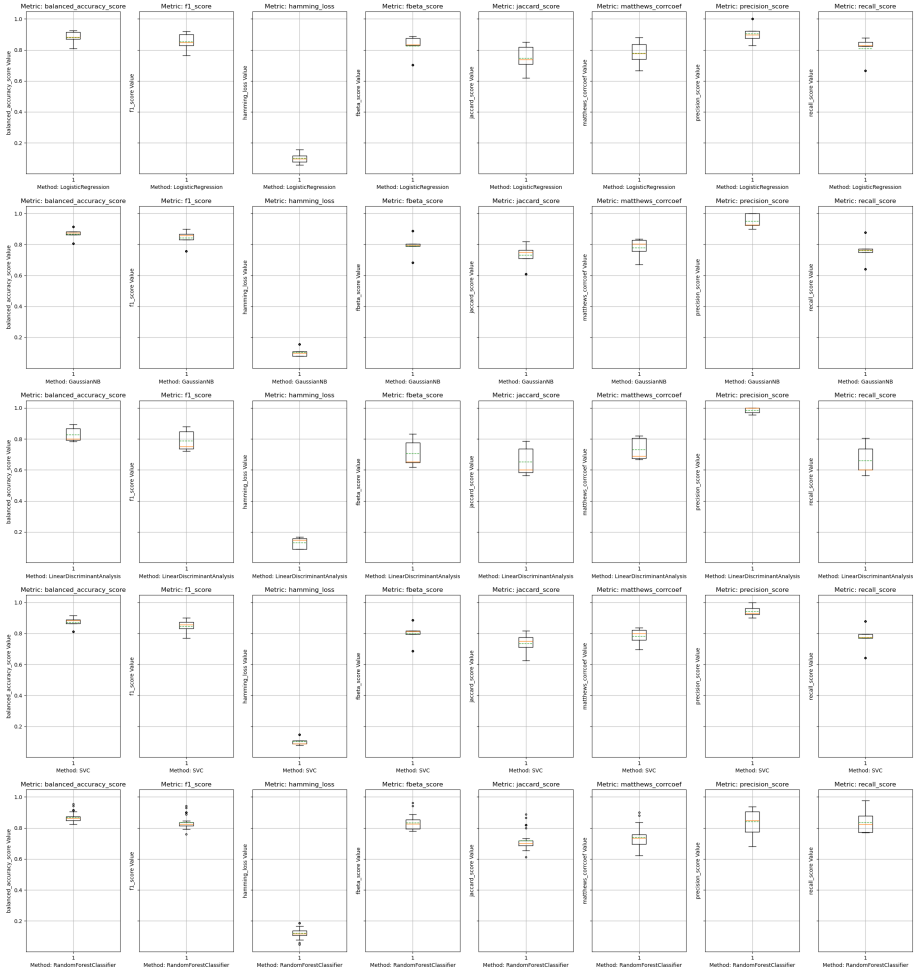

Figure 6: Final distributions of all loops of the feature selected model

We notice that the metrics do not improve much since there is little space to improve from the Baseline 3.2. Infact we expect that since we prune so many of our features to only remain with 2 form the initial 31 our metrics would decrease, here we see a resilience to this and a very good explanatory feature in our data.

3.4 Bonus 1

The previous feature selection method was a PCA, now we will try Select k best with a metric of X^2 and compare it to the previous feature selection however we will not take it into account for our winner model and the final conclusions to be fair.

Figure 7: Final distributions of all loops of the feature selected model using X^2 Select k best

Figure 8: Final distributions of all loops of the feature selected model using Mutual Information Select k best

In comparison to the PCA this feature selection has performed better, the assumption is that since the principled component analysis needs z-scaling before hand and we have not done this that it has performed worse. The Mutual Information infact has metrics that reach 1 which is most likely overfitting than anything else.

3.5 Optimized

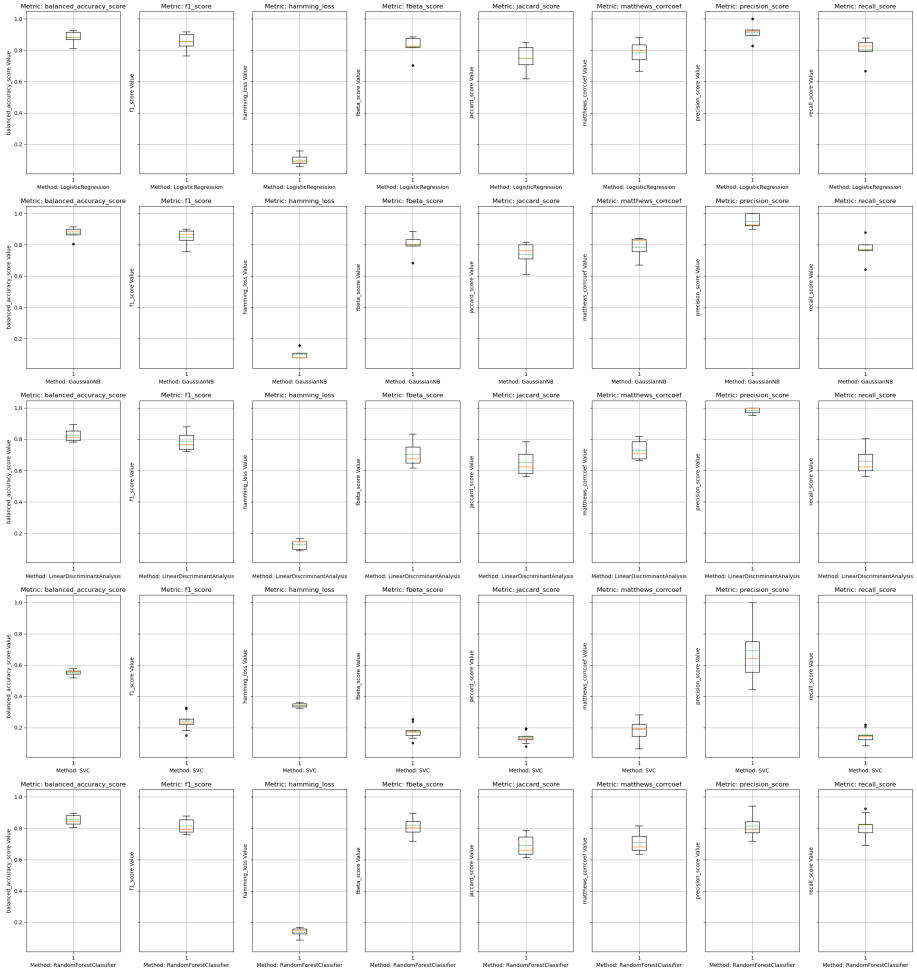

Figure 9: Final distributions of all loops of the optimized model

We can see that in our attempt to optimize for the fbeta score some of our models worsen quite alot for all metrics. We can see some marginal improvements on the fbeta score for some of the models (the mean and median value may remain the same but the distribution has leaned higher). The shape of the distributions of most of the models is similar to before. We can be sure that the SVC is not fit for this task. Keeping in mind that this optimized model uses the same feature selection as before we can see many improvements on the fbeta score and some other metrics for some of our models (for instance the Gaussian Naive Bayes and some of the metrics of the Linear Discriminant Analysis).

4 Conclusions

Our best models have to be between the Random Forest classifier, the Gaussian

Naive Bayes classifier and the Logistic Regression. We will train our winner on the whole dataset with the optimized parameter of one of these models, the Logistic Regression since it boasts one of the best fbeta scores that we care most for while maintaining very high metrics for the rest of the metrics (or low for the Hamming loss) and in particular the recall score which we care much for. This classification task an be solved well, to a high degree of satisfaction with relatively low amounts of data for the proper classification of both precision and recall.

5 Disclaimer

Large language models (LLM's) were used during the assignment.

5.1 LLM

Open AI: Chat GPT 4.0 [10]

5.2 Purpose

1. To ask if concepts already exist in some dependency.
2. For acquisition of the relative documentation.
3. For helping with the LaTeXtable formatting in this report for time efficiency.
4. Debugging help
5. In the function "plotKfoldSummary" to help with getting the box plots of all folds into one axis since 5 different box plots in each axis was generated until the .flatten method was called
6. For help when saving the winner since the models were ran but the winner was not saved so in order to save the winner we had to load back in a model from joblib but new methods had been added to the class. The help was to ask if when we load back in a model the new methods will be brought in, apparently they do since the pkl file keeps the attribute values and re calls the class upon loading from joblib with the parameters so it inherits the new methods.

References

[1] J. W. Koten and W. Den Otter, "The transition of benign to malignant in epithelial and mesenchymal tumours," *Anticancer Research*, vol. 11, no. 2, pp. 567–568, 1991, ISSN: 0250-7005.

[2] J. W. Koten, J. P. Neijt, B. A. Zonnenberg, and W. Den Otter, "The difference between benign and malignant tumours explained with the 4-mutation paradigm for carcinogenesis," *Anticancer Research*, vol. 13, no. 4, pp. 1179–1182, 1993, ISSN: 0250-7005.

[3] N. M. U. Din, R. A. Dar, M. Rasool, and A. Assad, "Breast cancer detection using deep learning: Datasets, methods, and challenges ahead," *Computers in Biology and Medicine*, vol. 149, p. 106 073, Oct. 2022, ISSN: 1879-0534. DOI: 10.1016/j.compbiomed.2022.106073.

[4] M. Radak, H. Y. Lafta, and H. Fallahi, "Machine learning and deep learning techniques for breast cancer diagnosis and classification: A comprehensive review of medical imaging studies," *Journal of Cancer Research and Clinical Oncology*, vol. 149, no. 12, pp. 10 473–10 491, Sep. 2023, ISSN: 1432-1335. DOI: 10.1007/s00432-023-04956-z.

[5] "3.13.2 documentation." (), [Online]. Available: https://docs.python.org/3/ (visited on 03/28/2025).

[6] "Scikit-learn: Machine learning in python — scikit-learn 1.6.1 documentation." (), [Online]. Available: https://scikit-learn.org/stable/ (visited on 03/28/2025).

[7] "Matplotlib documentation — matplotlib 3.10.1 documentation." (), [Online]. Available: https://matplotlib.org/stable/ (visited on 03/29/2025).

[8] "Optuna - a hyperparameter optimization framework," Optuna. (), [Online]. Available: https://optuna.org/ (visited on 03/29/2025).

[9] S. G. Kwak and J. H. Kim, "Central limit theorem: The cornerstone of modern statistics," *Korean Journal of Anesthesiology*, vol. 70, no. 2, pp. 144–156, Apr. 2017, ISSN: 2005-6419. DOI: 10.4097/kjae.2017.70.2.144.

[10] "ChatGPT." (), [Online]. Available: https://chatgpt.com (visited on 03/29/2025).