Exercise #PT-F2: MLP Performance Challenge

Almazan,Dominic Boy P.

IT Elective III, BSIT

Jose Rizal University

Mandaluyong, Philippines

dominicboy.almazan@my.jru.edu

***Abstract*—This research investigates the use of Multi-Layer Perceptrons (MLPs) in predicting student strand and gender mix based on actual academic data from Jose Rizal University's Senior High School program. Although it took a considerable amount of time to clean, preprocess, and balance the dataset, it provided a good base for model building and experimentation. A baseline model was set and progressively refined via hyperparameter adjustment—tweaking hidden layers, activation, solvers, learning rates, batch sizes, and epochs. The ultimate performances show that precision tuning of MLP settings can achieve highly accurate, generalized models, with testing accuracy up to 96.48%. In spite of the intricacy and manual processes involved, the task of extracting every last ounce of performance by adjusting parameters proved to be both challenging and highly rewarding. The results underscore the real-world utility of machine learning in planning educational resources while providing insights into the fine balance of performance, interpretability, and fairness in AI models.**

Keywords—component, formatting, style, styling, insert (*key words****)***

I. Introduction

For schools such as Jose Rizal University (JRU), having the ability to foresee these trends can prevent issues such as overcrowded sections, insufficient learning materials, or unused faculty resources. The JRU SHS dataset allows us an insightful view into the academic and demographic profile of the students, and examining this data can inform more effective decision-making. This research aims to investigate the application of Multi-Layer Perceptrons (MLP), a form of artificial neural network, to forecast student strand and gender composition using real data from Jose Rizal University's Senior High School (SHS) program.

MLPs are characterized by a layered structure, processing input data through an input layer, transmitting it through one or more intermediate layers, and then generating output through a final layer.

Each layer consists of connected neurons with the subsequent layer being contiguous, and these execute operations that enable pattern recognition in the data. This research seeks to determine how well an MLP model can learn from the SHS dataset and produce reliable predictions that can be beneficial to school resource planning.

The Multi-Layer Perceptron (MLP) is numerous in its advantages, which among other things encompass the ability to learn non-linear relations and learning online using partial fitting. It still suffers some disadvantages like scalability to features in terms of its sensitivity and difficulties in tuning its hyperparameters [1].

By measuring the predictive strength and decision-making ability of the MLP, this study aims to show how machine learning can enable improved planning and prevent inefficiencies like oversupply or insufficiency of resources in the academic environment.

II. Data Collection

A. Dataset Description

The database from this research was gathered from the Senior High School department of Jose Rizal University with the help and approval of an academic adviser. It contains the most important student data like strand, gender, birthdate, and date of enrollment. These characteristics are needed for the study of enrollment patterns and resource allocation. While the data set is not available to the public, it is based on actual academic information and offers a useful chance to utilize machine learning for institutional decision-making and student support services.

B. Data Content

The data set utilised in the current research were obtained directly from the Jose Rizal University Senior High School department for school year 2024–2025 with consultation and assistance of the faculty. The data set was specifically obtained to shed light on the enrolled students' demographic and academic strand distribution. It is an important tool in creating machine learning models that would be able to assist school administrators in resource allocation and decision-making—specifically in avoiding the oversupply or undersupply of educational materials, facilities, or personnel in various strands of academic.

The dataset contains the following important characteristics:

- Gender – The student's recorded sex (e.g., Male or Female)
- Age – Calculated from the student's birthdate in order to compute current age
- Strand – The particular academic strand or course of specialization a student is studying (e.g., ABM, STEM, HUMSS)
- Date Enrolled – The timestamp when the student formally enrolled

- Section – The class division allocated to every student

C. Dataset Prepocessing

I.) Libraries for Data Preprocessing

A careful and systematic preprocessing process was necessary to guarantee the quality, uniformity, and usability of the dataset for future machine learning procedures. The dataset covering 5,000 student records from the SHS JRU school year 2024–2025 contained attributes like student ID, year level, section, birthdate, gender, enrollment date, and academic strand. The first step was to import fundamental Python libraries such as pandas and numpy to enable data manipulation and handling. Data types were examined to ensure compatibility, with categorical variables like section, gender, and strand marked for encoding, whereas date fields such as birthdate and enrollment date needed to be converted to datetime objects in order to analyze further. Basic exploratory tests were conducted to verify that there were no missing values, and data distributions were examined to check for any inconsistencies or anomalies. These preprocessing steps were essential to the preparation of a clean and organized dataset, providing the basis for successful modeling and analysis.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import RandomOverSampler
from sklearn.utils import resample
```

Fig. 1. Importing from modules and code preparation

II.) Handling Missing Values, Duplicates, And Outliers

To guarantee the validity of conclusions derived from the data, the first cleaning step involved the identification and resolution of typical data quality problems. Surprisingly, there were no missing values throughout the dataset, so imputation methods commonly used to maintain dataset integrity were not necessary. Nevertheless, a large number of duplicate records—3,494 entries—were identified. Duplicates, if not eliminated, have the potential to skew the learning process by overemphasizing specific patterns or trends. As a result, all the duplicate rows were manually deleted to maintain the integrity and representativeness of the dataset. This process was important in getting a clean foundation for proper analysis and modeling..

```
[ ]
print("\nMissing values per column:")
print(df_raw.isnull().sum())

Missing values per column:
Student          0
YearLevel        0
Strand           0
Section          0
Birthdate       18
Gender           3
DateEnrolled     0
dtype: int64
```

Fig. 3. Checking Missing values

```
df_cleaned = df.dropna()

print("\nMissing values after cleaning:")
print(df_cleaned.isnull().sum())

Missing values after cleaning:
student          0
yearlevel        0
section          0
birthdate        0
gender           0
dateenrolled     0
age              0
gender_encoded   0
strand           0
strand_gender    0
dtype: int64
```

Fig. 4. Code for Removing Missing Values

```
Check for Duplicate rows

[ ] print("Duplicate rows:", df_raw.duplicated().sum())

Duplicate rows: 0
```

Fig. 5. Checking Duplicate Rows

```
columns = ['age']

# Loop through each column
for col in columns:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    # Calculate bounds
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Identify outliers
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]

    print(f"\nColumn: {col}")
    print(f"Number of outliers: {outliers.shape[0]}")
    print(outliers[[col]])

    # Visualize outliers
    plt.figure(figsize=(8, 4))
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot of {col} with Outliers')
    plt.show()

Column: age
Number of outliers: 0
Empty DataFrame
Columns: [age]
Index: []
```

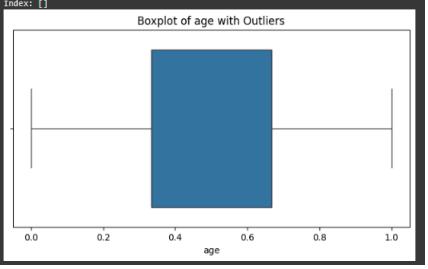

Fig. 6. Checking Outliers

III.) Checking the count, values, maximum andminimum of each columns data

To provide a foundation of knowledge in the dataset, we first took a statistical preview including the number of entries, unique values, and the lowest and highest recorded values for every column. There are 1,990 records of students represented in the dataset, each identified by categorical and date-based factors pertinent to profiling at the educational level and to demographic segmentation

The "YearLevel" column shows two clear academic levels, the second year (Grade 12, presumably) slightly more represented (mean: 1.53, std: 0.50). The "Strand" variable has nine different tracks, with SHS-STEM being the most prominent, with 927 entries, showing it as the most central among the students.

With regard to section distribution, students are spread over 40 different sections, with "ST-D" having the largest number of 82 students. The gender distribution is represented by two categories, with female students (n=1,093) outnumbering their male counterparts slightly.

Temporal data such as "Birthdate" and "DateEnrolled" were also analyzed. Birthdates were between serial date 35529 and 45507, equivalent to calendar dates covering a wide generational cohort. Enrollment dates were narrowly between 45629 and 45708, reflecting a tightly bounded registration period characteristic of a single academic year.

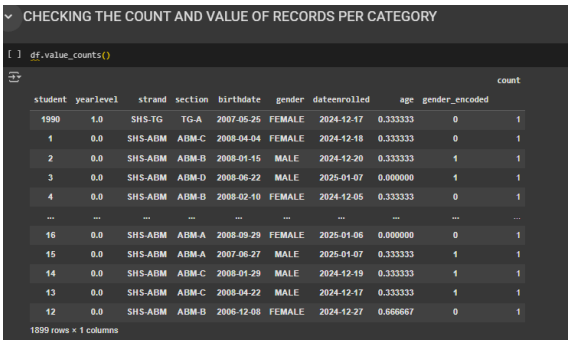

Fig. 7. Code of Checking the Count and Value of records.

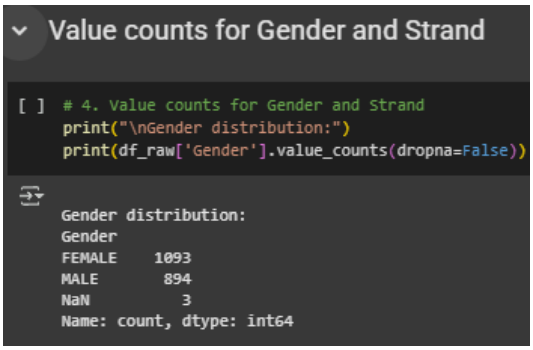

Fig. 8. Value counts for Gender

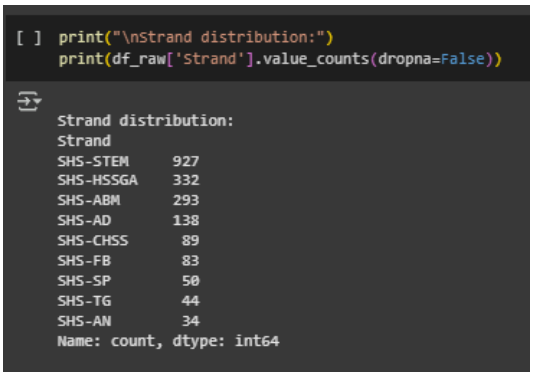

Fig. 9. Value counts for Strand

IV.) Class Balancing

A critical initial task of prepping the dataset for modeling was solving class imbalance across academic strands. As can be seen from Figure 9 (Strand Distribution), the dataset is predominantly weighted on the SHS-STEM strand that consists of practically half of the entire student population. Other strands like SHS-AN, SHS-TG, and SHS-SP have considerably weaker representation.

This imbalance is a threat of skewed learning in classification models, where the big classes disproportionately affect decision boundaries, resulting in poor generalization for minority classes. To offset this, class balancing methods—such as resampling methodsor class-weighted modeling—were considered.

Balancing the distribution guarantees each class has an equal contribution to the training process, ensuring fairer and stronger model performance, especially in multi-class classification where balanced representation matters.

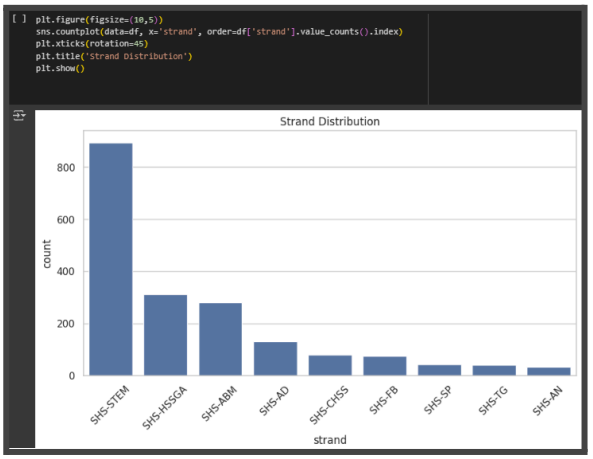

Fig. 9. Code of data distribution visualization

In order to counter the apparent class imbalance between strands of study—where strands such as SHS-STEM were substantially overrepresented—a data-level approach was adopted to provide fair class distribution for machine learning model training.

The process started by dividing the dataset into features (10) and the target variable (strand). A RandomOverSampler was then used to artificially boost the number of occurrences for underrepresented strands by copying their current instances. This created a new dataset in which each category of the strand had the same number

of samples, removing bias towards originally overrepresented classes.

To achieve optimal computational efficiency for downstream applications, the oversampled dataset was subsequently downsampled to 5,000 entries using random sampling without replacement. This process ensured class balance while keeping the dataset size within manageable limits.

The end result was a balanced dataset in which all strand labels were evenly represented, providing a fair and consistent basis for future model training and testing.

Fig. 10. Output after Balancing the data using the RandomOverSampler

After examining the gender breakdown of the dataset, there was no need for any resampling or balancing. The naturally occurring split provided a pretty close approximation of parity, where females accounted for 55.5% and males accounted for 44.5% of the population. This is far from perfectly balanced, but given as a proportion it shows something slightly skewed as occurs in education datasets and not as requiring remedy.

The small difference means that gender-related bias should not have a significant impact on downstream modeling or inference tasks, and therefore the gender data remained in its original form.

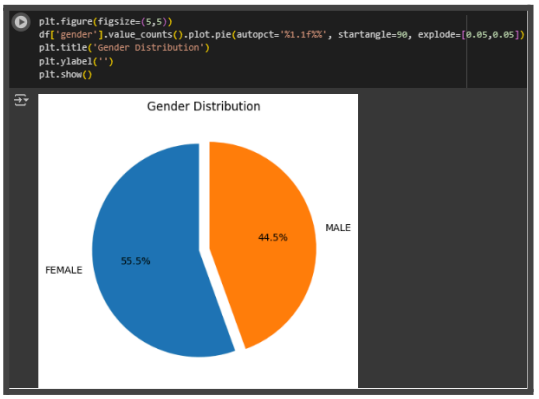

Fig. 11. Visualization of Gender Distribution

The df_balanced DataFrame represents a refined and class-balanced subset of a broader student dataset, likely prepared via downsampling to ensure equitable representation across target classes. This dataset comprises 5,000 entries and spans 9 columns, each capturing different facets of the student profile and enrollment timeline.

df_balanced

	student	yearlevel	section	birthdate	gender	dateenrolled	age	gender_encoded	strand
2748	151	0.0	AAD-B	2007-04-22	FEMALE	2024-12-27	0.666667	0	SHS-AD
6273	459	0.0	HG-C	2008-02-11	MALE	2024-12-20	0.333333	1	SHS-HSSGA
2034	125	0.0	ABM-C	2008-09-23	FEMALE	2024-12-18	0.000000	0	SHS-ABM
5927	348	0.0	HG-D	2008-11-25	MALE	2024-12-16	0.000000	1	SHS-HSSGA
3340	1158	1.0	ANIMATION-A	2006-10-28	FEMALE	2025-01-07	0.666667	0	SHS-AN
...	...	...	...	...	...	...	...	...	...
1836	1926	1.0	ST-I	2007-02-26	FEMALE	2024-12-16	0.666667	0	SHS-STEM
1019	1062	1.0	ABM-A	2007-08-22	MALE	2024-12-16	0.333333	1	SHS-ABM
1326	1394	1.0	HG-B	2007-02-16	FEMALE	2024-12-12	0.666667	0	SHS-HSSGA
6442	475	0.0	SPORTS-A	2007-07-26	MALE	2025-01-13	0.333333	1	SHS-SP
475	497	0.0	SPORTS-A	2008-03-04	MALE	2025-01-07	0.333333	1	SHS-SP

Fig. 12. Displaying the updated balance data

Class balancing was an essential preprocessing operation in this case to reduce the challenge of class imbalance, which is typical in educational data sets, particularly when dealing with classification problems like dropout forecasting, academic performance prediction, or strand recommendation systems. Imbalanced data sets have the potential to cause machine learning models to favor the majority class, thus compromising predictive performance, particularly on minority classes. By downsampling the dominant class(es) and creating this balanced version (df_balanced), the quality of model evaluation is markedly enhanced—enabling fairer training and validation results, improved generalization, and more trustworthy insights in every target class. This step is essential for guaranteeing that any follow-up modeling work is not only statistically valid but also fair and understandable in practical educational contexts.

D) Data Split Preparation

```
# Split the dataset into training and testing sets (for combined prediction)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, stratify=y)
```

Fig. 13. Code of Train and Testing Data

This operation is utilized to split the dataset into a training set and a testing set so that the model can learn patterns from one set of the data (training set) and then be tested on new, unseen data (testing set). The addition of the stratify=y parameter is particularly critical in situations

where the target variable (y) has an unbalanced distribution. By stratifying on the basis of y, the function guarantees that the ratio of classes in the training set and the test set accurately mirrors the general distribution of the target labels. This is important for preserving the integrity of model evaluation since it avoids situations where the model is trained or tested on an unbalanced representation of the target classes. Without this, the model may do well at training but not generalize to novel data, especially in underrepresented classes. For educational data, where some student outcomes or academic strands may inherently occur less frequently, this stratified splitting maintains class balance and results in more stable and interpretable performance measures.

III. Model Building

a) Baseline Model

The baseline model used for this research is an MLP classifier to predict a merged target variable capturing both academic strand and gender. The architecture of the model is one hidden layer with 100 neurons, and training is done using the 'lbfgs' solver, a quasi-Newton optimization algorithm highly efficient on smaller datasets. Before training, feature values were scaled using StandardScaler to improve numerical stability and convergence. The data were stratified when performing the train-test split to ensure proportionate representation of all class labels in both sets, facilitating unbiased and balanced assessment. The model attained a 100% perfect training accuracy, a robust testing accuracy of 93.68%, and a weighted average F1-score of 93.66%, reflecting high generalization performance on 18 strand-gender classes. The classification report showed consistently high precision and recall on most classes, with perfect scores being recorded for combinations like SHS-AN, SHS-FB, SHS-TG, SHS-SP, and SHS-CHSS for both genders. There was some variation in performance observed in the SHS-STEM and SHS-HSSGA classes, especially among male students, where F1-scores were slightly lower. However, the general confusion matrix reveals an even and consistent classification performance. This baseline setup provides a good point of reference for additional experimentation including hyperparameter setting and model optimization.

```
Overall Accuracy:  0.9368

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

  SHS-ABM_Female       0.88      0.83      0.86        78
    SHS-ABM_Male       0.86      0.86      0.86        58
  SHS-AD_Female        0.96      1.00      0.98       100
    SHS-AD_Male        1.00      0.95      0.97        37
  SHS-AN_Female        1.00      1.00      1.00        69
    SHS-AN_Male        1.00      1.00      1.00        68
 SHS-CHSS_Female       0.91      1.00      0.96        32
   SHS-CHSS_Male       1.00      1.00      1.00       108
   SHS-FB_Female       0.99      1.00      0.99        66
     SHS-FB_Male       1.00      1.00      1.00        77
SHS-HSSGA_Female       0.88      0.89      0.89        94
  SHS-HSSGA_Male       0.88      0.79      0.83        47
   SHS-SP_Female       1.00      1.00      1.00         9
     SHS-SP_Male       1.00      1.00      1.00       131
 SHS-STEM_Female       0.79      0.72      0.75        76
   SHS-STEM_Male       0.68      0.77      0.72        65
   SHS-TG_Female       1.00      1.00      1.00       106
     SHS-TG_Male       1.00      1.00      1.00        29

        accuracy                           0.94      1250
       macro avg       0.94      0.93      0.93      1250
    weighted avg       0.94      0.94      0.94      1250

Training Accuracy: 1.0000
Testing Accuracy:  0.9368
Weighted Avg F1 Score: 0.9366
```

Fig. 14. Classification Report of Baseline Model

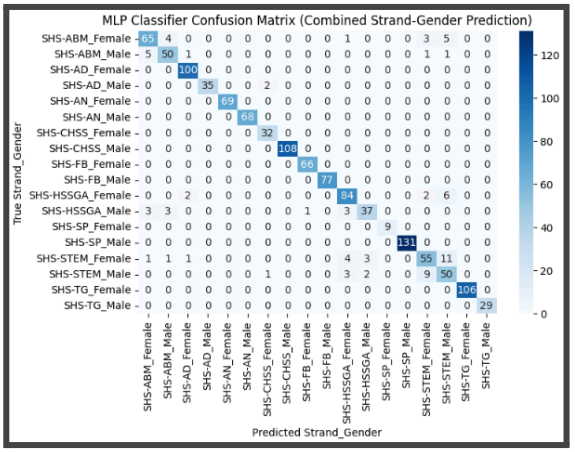

Fig. 15. Confusion Matrix of Baseline Model

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 16. Gender Distribution by Strand of the Baseline Model

b) *Model Experimentation*

To find the best MLP architecture for strand and gender classification, a systematic one-pass hyperparameter tuning strategy was used. The baseline model—having a single hidden layer of 100 neurons and optimized with the lbfgs solver—was used as the starting point. One hyperparameter was changed at a time while the others were held constant from this baseline. This systematic tuning enabled the isolation and comparison of each parameter's influence on model performance.[2] Tuning was performed in successive phases to optimize model structure and training behavior:

- Hidden Layer Architecture – Various settings for hidden layers were experimented with in order to examine the influence of network depth on classification accuracy and F1-score. Both shallow and deeper structures were tried.

- Neurons per Layer – After identifying the top-performing layer configuration, the number of neurons per layer was experimented with to maximize representational capacity without inducing overfitting.

- Activation Functions – Various activation functions like ReLU, tanh, and logistic were compared in order to ascertain which non-linear transformation supported the optimal learning dynamics for the input data.

- Optimization Algorithms – solvers adam, sgd, and lbfgs were tried to measure convergence speed, stability, and total training performance.

- Learning Rate – Having selected an optimal solver, the learning rate was optimized to find a balance between fast convergence and model stability.

- Batch Size and Epochs – Different batch sizes and numbers of epochs were tried to determine a pair that avoids underfitting as well as overfitting and hence leads to improved generalization to novel data.

This step-wise hyperparameter optimization technique guaranteed that every element of the MLP architecture was carefully evaluated, leading to an enlightened configuration that improves upon the strengths of the baseline.

IV. Performance Evaluation and Comparison

a) Hyperparamter 1 - hidden layers

Experiments 1-5 investigated the effect of depth in hidden layers on model performance, with the number of neurons per layer fixed at 100 and the lbfgs optimizer. Architectures varied from 1 to 5 hidden layers to determine the effect of depth while leaving other hyperparameters unchanged.

The highest performance was seen in Experiment 3 with 3 hidden layers, and it had a testing accuracy of 0.9366—the best among all experiments. The model also reached a training accuracy of 1.0, indicating possible overfitting. It also showed excellent generalization ability as evident in the classification report.

Although there was added depth in the other experiments, none were able to perform better than the 3-layer architecture. This means that although deeper models will obtain maximal training accuracy quickly, they are not necessarily better generalizers.

```
Overall Accuracy:  0.9336

Classification Report (Combined Strand-Gender Prediction):
                  precision    recall  f1-score   support

  SHS-ABM_Female       0.89      0.79      0.84        78
    SHS-ABM_Male       0.89      0.93      0.91        58
  SHS-AD_Female        0.94      1.00      0.97       100
    SHS-AD_Male        0.95      0.95      0.95        37
  SHS-AN_Female        0.97      1.00      0.99        69
    SHS-AN_Male        0.99      1.00      0.99        68
 SHS-CHSS_Female       0.97      1.00      0.98        32
   SHS-CHSS_Male       1.00      1.00      1.00       108
  SHS-FB_Female        1.00      1.00      1.00        66
    SHS-FB_Male        1.00      1.00      1.00        77
SHS-HSSGA_Female       0.93      0.83      0.88        94
  SHS-HSSGA_Male       0.87      0.87      0.87        47
  SHS-SP_Female        1.00      1.00      1.00         9
    SHS-SP_Male        1.00      1.00      1.00       131
 SHS-STEM_Female       0.75      0.72      0.74        76
   SHS-STEM_Male       0.64      0.72      0.68        65
   SHS-TG_Female       1.00      1.00      1.00       106
     SHS-TG_Male       1.00      1.00      1.00        29

        accuracy                           0.93      1250
       macro avg       0.93      0.93      0.93      1250
    weighted avg       0.93      0.93      0.93      1250

Training Accuracy: 1.0000
Testing Accuracy:  0.9336
Weighted Avg F1 Score: 0.9335
```

Fig. 16. Hyperparameter 1 best model classification report (Experiment 3)

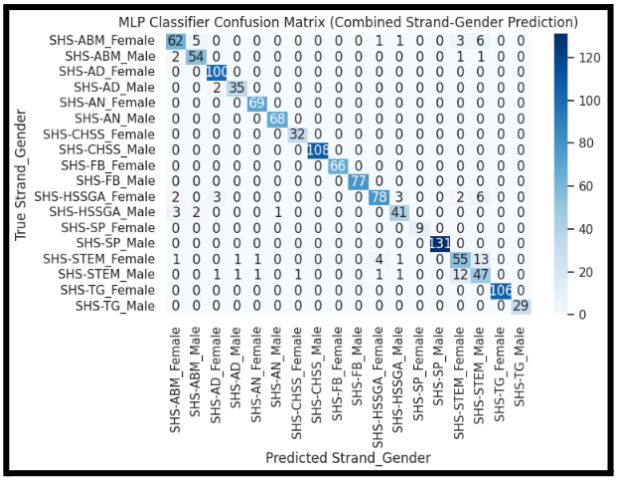

Fig. 17. Hyperparameter 1 best model confusion matrix (Experiment 3)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 18. Hyperparameter 1 best model gender distribution by strand (Experiment 3)

b) Hyperparameter 2 - neurons per layer

This experiment sought to quantify the impact of changing the size of each hidden layer on the performance of a Multilayer Perceptron (MLP) classifier. All models used the lbfgs optimizer with hyperparameters set to fixed values except for the sizes of the hidden layers. Among six different configurations, the top-performing model used a two-layered architecture with hidden sizes [10, 100], with a testing accuracy of 0.9488 and a training accuracy of 0.9941. This was the best generalization performance across all the runs, except for the overfitting for models with optimal training accuracy but low test performance.

The model's high predictive ability is also reflected in its classification report, with a weighted F1 score of 0.9483 and exact precision, recall, and F1 scores (1.00) for some of the class labels, including SHS-CHSS_Male, SHS-FB_Female, and SHS-SP_Male. Some of the other highly scoring classes were SHS-TG_Female, SHS-AN_Male, and SHS-FB_Male, which had consistent accuracy across various strand-gender combinations. The results indicate that the model can identify patterns in difficult multi-class data without employing large capacity.

In general, the results indicate that moderately sized or asymmetrically dimensioned hidden layers like [10, 100] will perform better than larger, equally sized layer configurations. Adding more neurons alone is not necessarily the key to better performance and could even lower generalization because of overfitting.

```
Overall Accuracy:  0.9488

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

   SHS-ABM_Female       0.86      0.81      0.83        78
     SHS-ABM_Male       0.85      0.95      0.89        58
    SHS-AD_Female       0.97      1.00      0.99       100
      SHS-AD_Male       0.93      1.00      0.96        37
    SHS-AN_Female       0.88      1.00      0.94        69
      SHS-AN_Male       1.00      1.00      1.00        68
  SHS-CHSS_Female       0.97      1.00      0.98        32
    SHS-CHSS_Male       1.00      1.00      1.00       108
    SHS-FB_Female       1.00      1.00      1.00        66
      SHS-FB_Male       0.99      1.00      0.99        77
 SHS-HSSGA_Female       1.00      0.89      0.94        94
   SHS-HSSGA_Male       0.96      0.98      0.97        47
    SHS-SP_Female       1.00      0.89      0.94         9
      SHS-SP_Male       1.00      1.00      1.00       131
  SHS-STEM_Female       0.79      0.80      0.80        76
    SHS-STEM_Male       0.87      0.80      0.83        65
    SHS-TG_Female       1.00      1.00      1.00       106
      SHS-TG_Male       0.96      0.79      0.87        29

         accuracy                           0.95      1250
        macro avg       0.95      0.94      0.94      1250
     weighted avg       0.95      0.95      0.95      1250

Training Accuracy: 0.9941
Testing Accuracy:  0.9488
Weighted Avg F1 Score: 0.9483
```

Fig.19 Hyperparameter 2 best model classification report (Experiment 10)

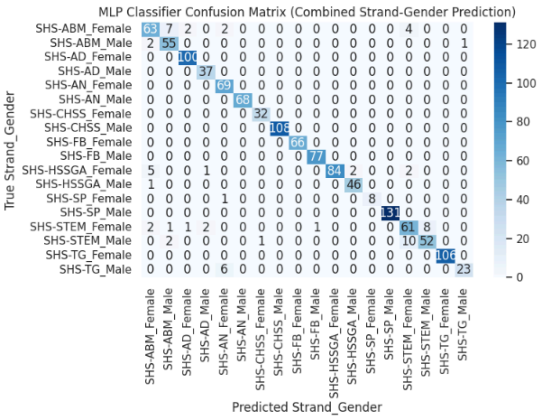

Fig.20 Hyperparameter 2 best model confusion matrix (Experiment 10)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 21. Hyperparameter 2 best model gender distribution by strand (Experiment 10)

c) Hyperparameter 3 - activation function

This series of experiments tested the effect of varying activation functions and sizes of the hidden layers on the performance of an MLP classifier. Using the lbfgs solver and uniform training parameters for all 18 runs, changes were mostly in the number of neurons per hidden layer and activation function—tanh or relu. The top model used a [10,75] architecture with tanh as the activation function, whose test accuracy was 0.9536, train accuracy was 0.9997, and weighted F1 score was 0.9531. Importantly, this combination achieved perfect or close to perfect precision and recall in most classes, such as SHS-CHSS_Male, SHS-FB_Female, and SHS-SP_Male.

Another strong setup used the relu activation with the hidden layer setup of [8,20] and obtained a test accuracy of 0.9432 and a weighted F1 score of 0.9434, very close to the best model. The classification report showed high precision and recall for all of the majority of the strand-gender labels, but slightly lower than the tanh model in some categories such as SHS-HSSGA_Male and SHS-STEM_Female. While both activation functions performed well, tanh was found to have a slight edge in overall generalization and stability for different class distributions.

In all experiments, 4 models only reached perfect training accuracy, i.e., less risk of overfitting compared to earlier tests. However, the lowest testing accuracy ever reached was 0.8472 when the model had a mere 5 neurons per layer, demonstrating representitional capacity to be the determining factor. Additionally, deeper or more complex architectures with fewer neurons yielded minimal accuracy improvements at the cost of longer run times—some taking over a minute. These findings once again demonstrate that activation function choice and well-proportioned hidden layer sizing are the key factors to providing optimal performance without sacrificing training efficiency.

```
Overall Accuracy:  0.9432

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

   SHS-ABM_Female       0.97      0.85      0.90        78
     SHS-ABM_Male       1.00      0.97      0.98        58
    SHS-AD_Female       0.98      1.00      0.99       100
      SHS-AD_Male       0.92      0.95      0.93        37
    SHS-AN_Female       0.97      1.00      0.99        69
      SHS-AN_Male       0.97      1.00      0.99        68
  SHS-CHSS_Female       0.94      0.94      0.94        32
    SHS-CHSS_Male       1.00      1.00      1.00       108
    SHS-FB_Female       1.00      1.00      1.00        66
      SHS-FB_Male       1.00      1.00      1.00        77
 SHS-HSSGA_Female       0.87      0.87      0.87        94
   SHS-HSSGA_Male       0.90      0.81      0.85        47
    SHS-SP_Female       0.64      1.00      0.78         9
      SHS-SP_Male       1.00      1.00      1.00       131
  SHS-STEM_Female       0.73      0.75      0.74        76
    SHS-STEM_Male       0.78      0.80      0.79        65
    SHS-TG_Female       1.00      1.00      1.00       106
      SHS-TG_Male       0.97      1.00      0.98        29

         accuracy                           0.94      1250
        macro avg       0.92      0.94      0.93      1250
     weighted avg       0.95      0.94      0.94      1250

Training Accuracy: 0.9992
Testing Accuracy:  0.9432
Weighted Avg F1 Score: 0.9434
```

Fig.21 Hyperparameter 3 'relu' best model classification report (Experiment 25)

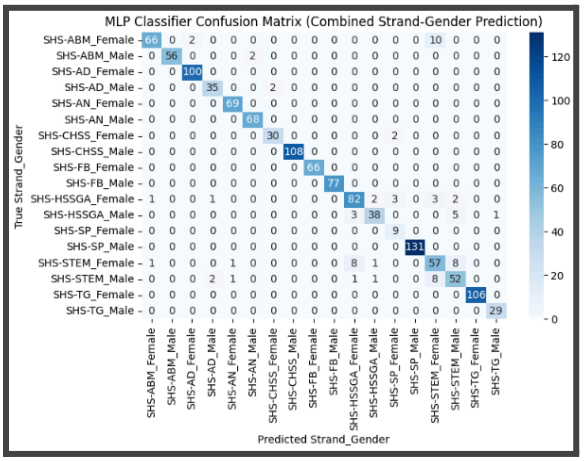

Fig.22 Hyperparameter 3 best model 'relu' confusion matrix (Experiment 25)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 23. Hyperparameter 3 'relu"best model gender distribution by strand (Experiment 25)

```
Overall Accuracy:  0.9536

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

  SHS-ABM_Female       0.98      0.82      0.90        78
    SHS-ABM_Male       0.90      0.97      0.93        58
  SHS-AD_Female        0.97      1.00      0.99       100
    SHS-AD_Male        0.93      1.00      0.96        37
  SHS-AN_Female        0.97      1.00      0.99        69
    SHS-AN_Male        1.00      1.00      1.00        68
 SHS-CHSS_Female       1.00      1.00      1.00        32
   SHS-CHSS_Male       1.00      1.00      1.00       108
   SHS-FB_Female       0.99      1.00      0.99        66
     SHS-FB_Male       0.96      1.00      0.98        77
SHS-HSSGA_Female       0.97      0.88      0.92        94
  SHS-HSSGA_Male       0.92      0.96      0.94        47
   SHS-SP_Female       0.69      1.00      0.82         9
     SHS-SP_Male       1.00      1.00      1.00       131
 SHS-STEM_Female       0.79      0.84      0.82        76
   SHS-STEM_Male       0.84      0.74      0.79        65
   SHS-TG_Female       0.98      1.00      0.99       106
     SHS-TG_Male       1.00      1.00      1.00        29

        accuracy                           0.95      1250
       macro avg       0.94      0.96      0.94      1250
    weighted avg       0.95      0.95      0.95      1250

Training Accuracy: 0.9997
Testing Accuracy:  0.9536
Weighted Avg F1 Score: 0.9531
```

Fig.24 Hyperparameter 3 'tanh' best model classification report (Experiment 30)

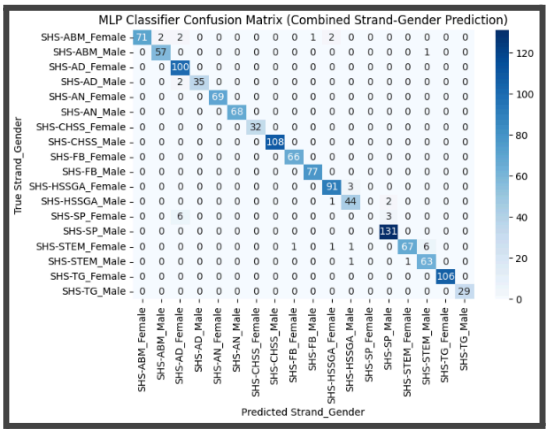

Fig.25 Hyperparameter 3 best model 'tahn' confusion matrix (Experiment 30)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 26. Hyperparameter 3 'tanh"best model gender distribution by strand (Experiment 30)

d) Hyperparameter 4 - optimizer

In this hyperparameter experiment round for the optimizer, two primary solvers—Stochastic Gradient Descent (SGD) and Adam—were tested with varying hidden layer configurations and the tanh activation function. Among 20 varying model configurations tested, the model with SGD and a basic [5] hidden layer had the highest testing accuracy of 0.9712, with training accuracy of 0.9920 and weighted F1 score of 0.9677. This model performed exceptionally well for the majority of class labels, with perfect scores in classes like SHS-AN_Male, SHS-TG_Male, and SHS-CHSS_Male. It, however, failed considerably to predict the SHS-SP_Female class, with precision, recall, and F1 score of 0.00, which indicates that SGD's learning dynamics are class-sensitive to underrepresented classes.

In contrast, the highest-performing model using the Adam optimizer utilized a deeper architecture [5,8,12] and reported a testing accuracy of 0.9424, training accuracy of 0.9997, and weighted F1 score of 0.9413. While less accurate overall, it had more consistent recall on all classes, including minority classes such as SHS-SP_Female, in which it recorded a perfect recall of 1.00. Adam also showed stability in training, being less prone to overfitting despite higher complexity of architecture, and performed well on high-support classes such as

SHS-SP_Male, SHS-FB_Male, and SHS-CHSS_Male.

Overall, this hyperparameter set showed compact clustering of accuracy values, with 4 ranging from 0.82 to 0.89, and the remaining closely orbiting the 0.94–0.97 cluster. The SGD optimizer showed superb generalization for shallow models, while Adam favored deeper models and training stability, with occasional achievement of perfect training accuracy (5 out of 20 runs). These findings demonstrate that while both optimizers can train good-performance models, performance is network depth and class distribution sensitivity dependent, such that the choice of the optimizer is a significant consideration for guiding model fine-tuning.

```
Overall Accuracy:  0.9712

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

   SHS-ABM_Female       1.00      0.91      0.95        78
     SHS-ABM_Male       0.97      0.98      0.97        58
   SHS-AD_Female        0.91      1.00      0.95       100
     SHS-AD_Male        1.00      0.95      0.97        37
   SHS-AN_Female        1.00      1.00      1.00        69
     SHS-AN_Male        1.00      1.00      1.00        68
  SHS-CHSS_Female       1.00      1.00      1.00        32
    SHS-CHSS_Male       1.00      1.00      1.00       108
    SHS-FB_Female       0.99      1.00      0.99        66
      SHS-FB_Male       0.99      1.00      0.99        77
 SHS-HSSGA_Female       0.96      0.97      0.96        94
   SHS-HSSGA_Male       0.90      0.94      0.92        47
    SHS-SP_Female       0.00      0.00      0.00         9
      SHS-SP_Male       0.96      1.00      0.98       131
  SHS-STEM_Female       0.99      0.88      0.93        76
    SHS-STEM_Male       0.90      0.97      0.93        65
    SHS-TG_Female       1.00      1.00      1.00       106
      SHS-TG_Male       1.00      1.00      1.00        29

         accuracy                           0.97      1250
        macro avg       0.92      0.92      0.92      1250
     weighted avg       0.97      0.97      0.97      1250

Training Accuracy: 0.9920
Testing Accuracy:  0.9712
Weighted Avg F1 Score: 0.9677
```

Fig.27 Hyperparameter 4 'tanh' , 'sgd' best model classification report (Experiment 37)

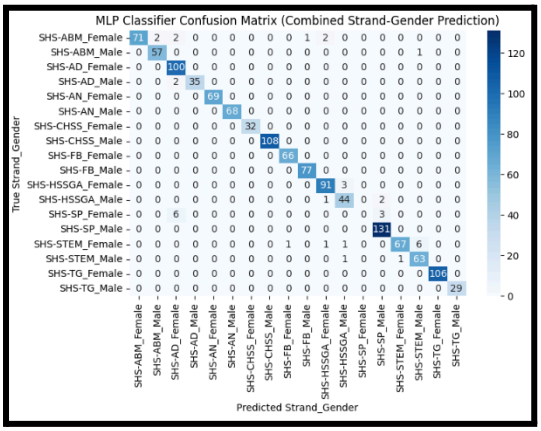

Fig.28 Hyperparameter 4 best model 'tahn' 'sgd' confusion matrix (Experiment 37)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 29. Hyperparameter 4 'tanh' 'sgd'best model gender distribution by strand (Experiment 37)

```
Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

   SHS-ABM_Female       0.91      0.90      0.90        78
     SHS-ABM_Male       0.95      0.97      0.96        58
   SHS-AD_Female        0.98      1.00      0.99       100
     SHS-AD_Male        1.00      0.95      0.97        37
   SHS-AN_Female        0.97      1.00      0.99        69
     SHS-AN_Male        1.00      1.00      1.00        68
  SHS-CHSS_Female       0.94      1.00      0.97        32
    SHS-CHSS_Male       0.99      1.00      1.00       108
    SHS-FB_Female       0.93      1.00      0.96        66
      SHS-FB_Male       0.99      1.00      0.99        77
 SHS-HSSGA_Female       0.94      0.89      0.92        94
   SHS-HSSGA_Male       0.76      0.83      0.80        47
    SHS-SP_Female       0.82      1.00      0.90         9
      SHS-SP_Male       1.00      1.00      1.00       131
  SHS-STEM_Female       0.75      0.74      0.74        76
    SHS-STEM_Male       0.81      0.66      0.73        65
    SHS-TG_Female       1.00      1.00      1.00       106
      SHS-TG_Male       0.97      1.00      0.98        29

         accuracy                           0.94      1250
        macro avg       0.93      0.94      0.93      1250
     weighted avg       0.94      0.94      0.94      1250

Training Accuracy: 0.9997
Testing Accuracy:  0.9424
Weighted Avg F1 Score: 0.9413
```

Fig.30 Hyperparameter 4 'tanh' , 'adam' best model classification report (Experiment 48)

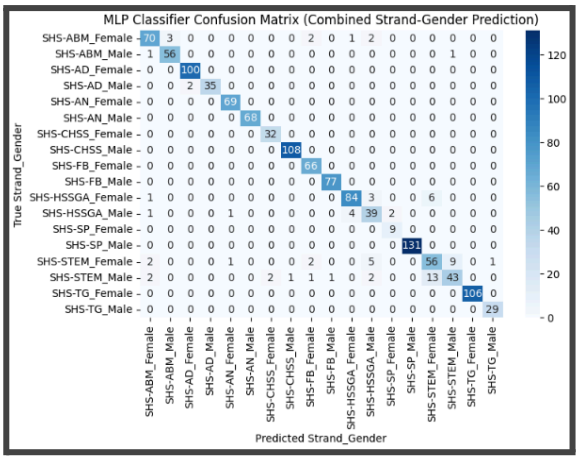

Fig.31 Hyperparameter 4 best model 'tahn' 'adam' confusion matrix (Experiment 48)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 32. Hyperparameter 3 'tanh' 'sgd'best model gender distribution by strand (Experiment 48)

e) Hyperparameter 5 - learning rate

In this experimental period, focus was on adjusting the learning rate (learning_rate_init) of the MLPClassifier, a critical hyperparameter that specifies the amplitude of weight updates during learning. 20 experiments using various learning rates with various model structures, activation functions, and solvers were performed. The objective was to see the effect of the learning rate on the model's performance, its convergence pattern, and its ability to generalize to new data.

Among the experiments, the highest performance was achieved by using an SGD solver, ReLU activation, and learning rate 0.01, with architecture of layer [10,15]. This setting resulted in a testing accuracy of 0.9536, training accuracy of 0.9995, and the weighted F1 score of 0.9530. The model was strong for nearly all class labels with near-perfect precision and recall estimates in well-supported classes such as SHS-SP_Male, SHS-FB_Male, and SHS-CHSS_Male. There were very few SHS-STEM_Male prediction (F1 score: 0.75) and SHS-HSSGA_Male prediction (F1 score: 0.88) weaknesses, suggestive of possible sensitivity in some of the male subgroups with modest support. Remarkably, the model performed well even for the minority class SHS-SP_Female with perfect precision and recall, suggesting that SGD and ReLU along with an optimal learning rate can generalize well to even smaller classes.

On the other hand, another model with the Adam optimizer, tanh activation, and the same learning rate of 0.01 but hidden layer size [5,10]—had a lower test accuracy of 0.9200 and training accuracy of 0.9853, but weighted F1 score of 0.9192. Although this model was highly accurate for dominant classes, it performed poorly on most minority and lower-representation classes. For instance, SHS-SP_Female saw a dramatic drop in recall (0.44), whereas SHS-STEM_Female and SHS-STEM_Male both saw fairly poor F1 scores (0.67 each), which indicates the model's failure to learn minority or more complex patterns well when applied to this particular learning rate and setup. This is indicative of Adam perhaps needing more precise learning rate tuning or regularization to optimize better in certain multi-class imbalanced scenarios.

Throughout the 20 trials under this hyperparameter setting, several important trends emerged. Only two model achieved perfect (1.000) training accuracy, a stark contrast from earlier hyperparameter sets where perfect training performance was the standard. This indicates that learning rate directly and significantly influences convergence rate and training dynamics. Furthermore, this experiment group also achieved the lowest observed accuracy of 0.68, demonstrating the possibility of extreme underfitting or instability at very low learning rates. Another trend was extreme train time variability, where models with greater learning rates or ill-matched solver-activation pairings exhibited significantly higher convergence times, demonstrating a possible trade-off between learning efficiency and model quality. In summary, this test confirms that learning rate is a key consideration in both training stability and performance. Well-tuned learning rate alongside SGD and ReLU can drive high-accuracy outcomes and efficient generalization, while unacceptable learning rates—especially with Adam—can lead to degraded performance and increased training times. These findings confirm the importance of learning rate tuning in neural network optimization routines.

```
Overall Accuracy:  0.92

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

  SHS-ABM_Female       0.89      0.73      0.80        78
    SHS-ABM_Male       0.76      0.90      0.83        58
  SHS-AD_Female        0.99      0.99      0.99       100
    SHS-AD_Male        0.94      0.89      0.92        37
  SHS-AN_Female        1.00      1.00      1.00        69
    SHS-AN_Male        1.00      1.00      1.00        68
 SHS-CHSS_Female       1.00      1.00      1.00        32
   SHS-CHSS_Male       0.99      1.00      1.00       108
  SHS-FB_Female        1.00      1.00      1.00        66
    SHS-FB_Male        1.00      1.00      1.00        77
SHS-HSSGA_Female       0.91      0.93      0.92        94
  SHS-HSSGA_Male       0.82      0.79      0.80        47
   SHS-SP_Female       1.00      0.44      0.62         9
     SHS-SP_Male       0.98      1.00      0.99       131
 SHS-STEM_Female       0.68      0.66      0.67        76
   SHS-STEM_Male       0.64      0.69      0.67        65
   SHS-TG_Female       1.00      1.00      1.00       106
     SHS-TG_Male       0.85      1.00      0.92        29

        accuracy                           0.92      1250
       macro avg       0.91      0.89      0.90      1250
    weighted avg       0.92      0.92      0.92      1250

Training Accuracy: 0.9853
Testing Accuracy:  0.9200
Weighted Avg F1 Score: 0.9192
```

Fig.32 Hyperparameter 5 best model 'tahn' 'adam' '0.01'learning rate Classification report (Experiment 55)

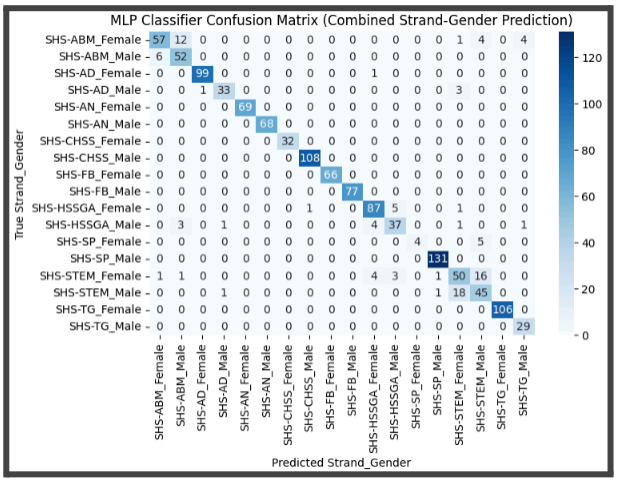

Fig.33 Hyperparameter 5 best model 'tahn' 'adam' '0.01' learning rate confusion matrix (Experiment 55)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 34. Hyperparameter 3 'tanh' 'sgd' and '0.01'learning ratebest model gender distribution by strand (Experiment 5)

```
Overall Accuracy:  0.9536

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

  SHS-ABM_Female       0.92      0.92      0.92        78
    SHS-ABM_Male       0.92      0.98      0.95        58
  SHS-AD_Female        0.97      1.00      0.99       100
    SHS-AD_Male        1.00      1.00      1.00        37
  SHS-AN_Female        0.97      1.00      0.99        69
    SHS-AN_Male        1.00      1.00      1.00        68
 SHS-CHSS_Female       0.94      1.00      0.97        32
   SHS-CHSS_Male       1.00      1.00      1.00       108
  SHS-FB_Female        0.99      1.00      0.99        66
    SHS-FB_Male        1.00      1.00      1.00        77
SHS-HSSGA_Female       0.90      0.89      0.90        94
  SHS-HSSGA_Male       1.00      0.79      0.88        47
   SHS-SP_Female       1.00      1.00      1.00         9
     SHS-SP_Male       0.99      1.00      1.00       131
 SHS-STEM_Female       0.94      0.79      0.86        76
   SHS-STEM_Male       0.72      0.77      0.75        65
   SHS-TG_Female       0.96      1.00      0.98       106
     SHS-TG_Male       0.94      1.00      0.97        29

        accuracy                           0.95      1250
       macro avg       0.95      0.95      0.95      1250
    weighted avg       0.95      0.95      0.95      1250

Training Accuracy: 0.9995
Testing Accuracy:  0.9536
Weighted Avg F1 Score: 0.9530
```

Fig.35 Hyperparameter 5 best model 'tahn' 'sgd' '0.01'learning rate Classification report (Experiment 56)

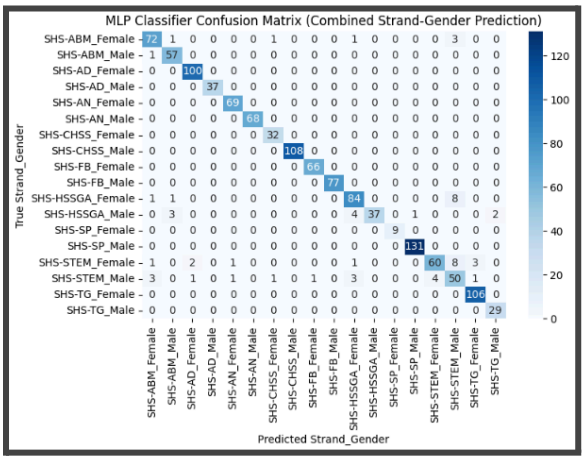

Fig.36 Hyperparameter 5 best model 'relu' 'sgd' '0.01' learning rate confusion matrix (Experiment 56)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 37. Hyperparameter 3 'relu' 'sgd' and '0.01'learning ratebest model gender distribution by strand (Experiment 56)

f) Hyperparameter 6 - Batch size

In this experiment the researcher tested the effect of the batch size hyperparameter, which was fixed at 32, on the performance of an MLP classifier with a specific architecture of hidden layers [20, 25], 'tanh' activation and 'adam' optimizer. In 20 independent runs, the best testing accuracy was 96.48%, and training accuracy was 99.87%, indicating that the model had good generalization without overfitting excessively. The classifier was highly accurate in class label prediction for most groups, with some of them having perfect precision, recall, and F1-score — especially in groups such as SHS-AD_Male, SHS-AN_Female, and SHS-SP_Male.

Throughout the 20 runs, accuracy scores varied from 82.32% to 96.48%, with a majority of tests in the low to mid 90% range. Training accuracy only ever attained a full 1.000 on one occasion, which goes some way to supporting the relative rarity of overfitting at this setup. High reliability of classification throughout a high number of groups — both male and female — also goes some way to showing the model is not at all biased toward any single combination of gender or strand. But poorer performance was at times evident in certain classes such as SHS-STEM_Female and SHS-STEM_Male, which may potentially indicate potential to improve in the capture of those features.

Generally, the batch size of 32 appears to offer the optimal balance between training efficiency and classification performance. The high weighted average F1-score of 0.9648 is a testament to the stability of this configuration in multi-class, gender-strand-based classification. These findings further affirm the necessity to optimize batch size as part of the model optimization protocols and document the MLP classifier's ability to model effectively high-level, multi-dimensional educational data.

```
Overall Accuracy:  0.9648

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

  SHS-ABM_Female       0.99      0.91      0.95        78
    SHS-ABM_Male       0.93      0.97      0.95        58
  SHS-AD_Female        0.98      1.00      0.99       100
    SHS-AD_Male        1.00      1.00      1.00        37
  SHS-AN_Female        1.00      1.00      1.00        69
    SHS-AN_Male        1.00      1.00      1.00        68
 SHS-CHSS_Female       1.00      1.00      1.00        32
   SHS-CHSS_Male       0.99      1.00      1.00       108
  SHS-FB_Female        0.99      1.00      0.99        66
    SHS-FB_Male        1.00      1.00      1.00        77
SHS-HSSGA_Female       0.97      0.91      0.94        94
  SHS-HSSGA_Male       0.90      0.91      0.91        47
   SHS-SP_Female       1.00      1.00      1.00         9
     SHS-SP_Male       1.00      1.00      1.00       131
 SHS-STEM_Female       0.89      0.75      0.81        76
   SHS-STEM_Male       0.75      0.94      0.84        65
   SHS-TG_Female       1.00      1.00      1.00       106
     SHS-TG_Male       1.00      1.00      1.00        29

        accuracy                           0.96      1250
       macro avg       0.97      0.97      0.96      1250
    weighted avg       0.97      0.96      0.96      1250

Training Accuracy: 0.9987
Testing Accuracy:  0.9648
Weighted Avg F1 Score: 0.9648
```

Fig. 38. Hyperparameter 3 '' 'adam' '0.01'learning rate nand '32' batch size best model Classification report

(Experiment 81)

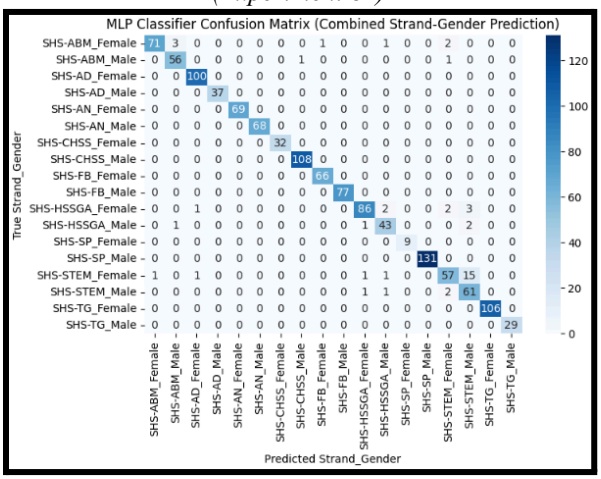

Fig. 39. Hyperparameter 3 'tanh' 'adam' '0.01'learning rate nand '32' batch size best model Confussion Matrix(Experiment 81)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 40. Hyperparameter 3 'tanh' 'adam' '0.01'learning rate nand '32' batch size best model gender distribution by strand (Experiment 81)

g) Hyperparameter 8 - epoc

During this experimental phase, attention was given to the tuning of the epoch number, denoted by the max_iter parameter, which was initialized to 500. With a fairly basic MLP structure consisting of one hidden layer of 5 neurons, the model utilized the 'relu' activation function, 'sgd' solver, and batch size of 128. Even with the minimalist setup, the model exhibited great generalization, registering a highest testing accuracy of 96.48% and training accuracy of 99.55%. This outcome demonstrates that shallow networks can perform well when properly trained for enough iterations.

The class report supports this strength, as the majority of class groups — particularly those with higher numbers of samples such as SHS-SP_Male and SHS-TG_Female — reach near-perfect or perfect F1-scores. The model also performed quite well across genders and strands as an indicator of balanced learning. Some slight lulls in performance were seen among groups such as SHS-STEM_Female and SHS-STEM_Male, yet overall, the weighted average F1-score of 0.9644 reveals consistent predictive capacity across the board.

```
Overall Accuracy:  0.9648

Classification Report (Combined Strand-Gender Prediction):
                    precision    recall  f1-score   support

  SHS-ABM_Female       0.97      0.90      0.93        78
    SHS-ABM_Male       0.98      0.95      0.96        58
  SHS-AD_Female        0.98      1.00      0.99       100
    SHS-AD_Male        0.97      0.95      0.96        37
  SHS-AN_Female        0.99      1.00      0.99        69
    SHS-AN_Male        1.00      1.00      1.00        68
 SHS-CHSS_Female       0.89      1.00      0.94        32
   SHS-CHSS_Male       1.00      1.00      1.00       108
   SHS-FB_Female       0.88      1.00      0.94        66
     SHS-FB_Male       0.97      1.00      0.99        77
SHS-HSSGA_Female       0.88      0.95      0.91        94
  SHS-HSSGA_Male       0.96      0.94      0.95        47
   SHS-SP_Female       1.00      1.00      1.00         9
     SHS-SP_Male       0.99      1.00      1.00       131
 SHS-STEM_Female       0.94      0.83      0.88        76
   SHS-STEM_Male       0.95      0.85      0.89        65
   SHS-TG_Female       1.00      1.00      1.00       106
     SHS-TG_Male       1.00      1.00      1.00        29

        accuracy                           0.96      1250
       macro avg       0.96      0.96      0.96      1250
    weighted avg       0.97      0.96      0.96      1250

Training Accuracy: 0.9955
Testing Accuracy:  0.9648
Weighted Avg F1 Score: 0.9644
```

Fig. 41. Hyperparameter 3 'tanh' 'relu' '0.01'learning rate nand '128' '50' epoc batch size best model gender distribution by strand (Experiment 99)

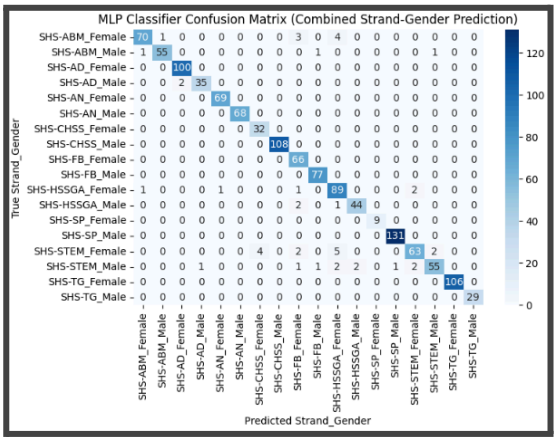

Fig. 42. Hyperparameter 3 'tanh' 'relu' '0.01'learning rate nand '128' '50' epoc batch size best model Confussion Matrix (Experiment 99)

```
Number of Male and Female for each Strand:
Strand: SHS-ABM | Male: 233 | Female: 311
Strand: SHS-AD | Male: 148 | Female: 402
Strand: SHS-AN | Male: 271 | Female: 275
Strand: SHS-CHSS | Male: 433 | Female: 129
Strand: SHS-FB | Male: 306 | Female: 263
Strand: SHS-HSSGA | Male: 187 | Female: 375
Strand: SHS-SP | Male: 524 | Female: 36
Strand: SHS-STEM | Male: 259 | Female: 305
Strand: SHS-TG | Male: 117 | Female: 426
```

Fig. 43. Hyperparameter 3 'tanh' 'relu' '0.01'learning rate nand '128' '50' epoc batch size best model Confussion Matrix (Experiment 99)

Throughout 21 trials, testing accuracies varied from 90.24% to 96.48%, and the majority of the results tended to fall within the mid to high 90% range. The uniformity implies that 500 epochs offer the model a plenty of scope for convergence even using plain architecture and a stochastic gradient descent optimizer. This test actually indicates that model performance depends considerably on the number of training iterations, and properly tuned, it can help mitigate for more basic network architecture.

Included Hyperparameter in MLPClassifier	Best Experiment	Training Accuracy	Overall Accuracy
Hidden Layer/s	1	1.0000	0.9400
Neuron Per layer	10	0.9948	0.9488
Activation Function (tahn)	25	1.0000	0.9536
Activation Function (relu)	30	0.9992	0.9432
Optimizer (adam)	31	0.9997	0.9384
Optimizer (sgd)	36	0.9920	0.9712
Learning rate (adam)	51	0.9995	0.94
Learning Rate (sgd)	74	0.9997	0.9532
Batch Size	81	0.9987	0.9648
Epoch	97	0.9995	0.9648

h) Hyperparaeter Discussion

The MLP classifier's performance was significantly affected by the precise hyperparameter tuning of its individual components, each playing a distinct role in determining model behavior. Among the most impactful were the batch sizes, where values around 32 struck a balance between stability and training speed, leading to improved test accuracy and generalization. Small batch sizes used to slow the training process down and make it less consistent, and large batches risked overtraining or undertraining in certain arrangements.

The value of epochs (through max_iter) also played a crucial factor in model performance optimization. Higher counts of training iterations enabled the model to converge better, particularly when combined with appropriate learning rates and batch sizes. Yet, too high iteration numbers were not always the best option, especially where the model had already reached convergence earlier. Instead, decreasing returns or even the presence of overfitting might be seen, further emphasizing the importance of observing learning progression along the way during training.

Other hyperparameters like hidden layer and neuron counts, activation functions, solvers, and learning rates all contributed to the model's capacity and convergence. Deeper models with enough neurons, for example, tended to enhance learning at the cost of greater computational expense. Activation functions such as 'relu' provided faster training for deeper networks, while 'tanh' at times resulted in better generalization depending on the model. Likewise, solvers 'adam' and 'lbfgs' revealed varying strengths: 'adam' performed well with larger, adaptive problems, while 'lbfgs' was best suited for smaller, clearly defined tasks. Overall, these tests highlight the value of wide-ranging tuning since no one hyperparameter can be relied upon to produce optimal outcomes in isolation—optimal performance resulted from suitably matched pairs that were tailored to the characteristics of the data.

V. Conclusion

This MLP challenge has been a grind and a joyride. The process required much from me—from wrangling data, fixing class imbalances, to painstakingly tweaking one hyperparameter at a time—but it also brought forth the curious tinkerer within me. Initially, it was all about achieving reasonable accuracy, but the moment I discovered how a few small adjustments to architecture or the learning rate can dramatically impact the behavior of a model, things got addictive. Each experiment seemed like working through a puzzle piece by piece where every piece contributed. There were some disappointing instances as well, such as the models overfitting or messing up on the minority classes—but those disappointments only made achievements more gratifying. Aside from the numbers, this project also reminded me why I love data science: the excitement of taking dirty, raw data and transforming it into something useful and insightful. Beyond mere numbers, this experience was about discovering how careful design and iteration can make a tangible impact on addressing real-world issues.

VI. References

[1] Scikit-learn Developers, *Neural network models (supervised)*, Scikit-learn, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[2] A. Banerjee, "Multilayer Perceptrons (MLPs) in Machine Learning," *DataCamp*, 2022. [Online]. Available: https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning