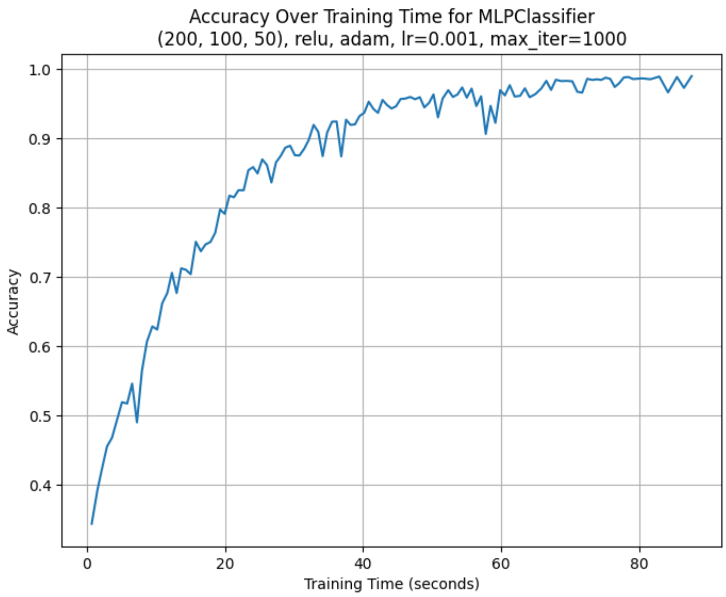
A. Changing the Network Structure, Transfer Functions, Learning Rate, Algorithm, and Stop Conditions

Introduction: In this report, I analyse the performance of various MLPClassifier configurations based on five tests conducted with different hyperparameters. I compare these configurations based on iteration number, graph fluctuations, final accuracy, and total training time.

Test Configurations:

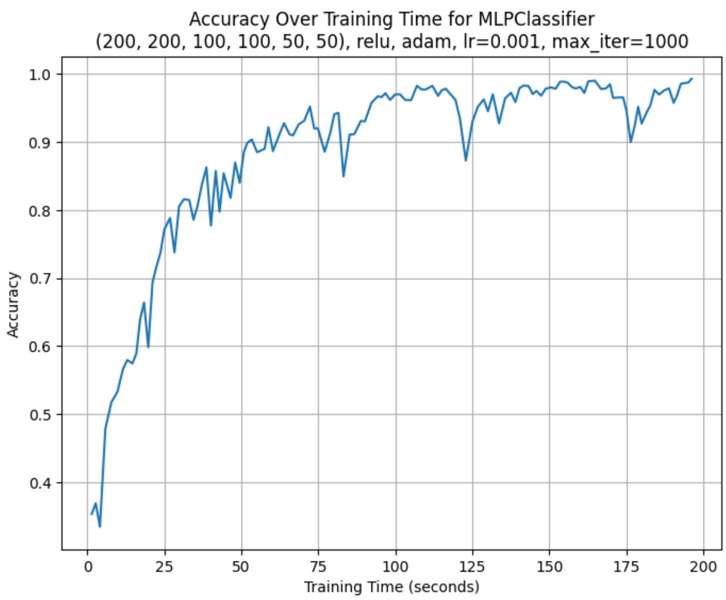
1. **Test 1:**

    - **Configuration:** Hidden Layer Sizes = (200, 100, 50), Activation = 'ReLU', Solver = 'Adam', Learning Rate = 0.001, Max Iter = 1000
    - **Final Accuracy:** 0.990
    - **Total Training Time:** 87.593 seconds



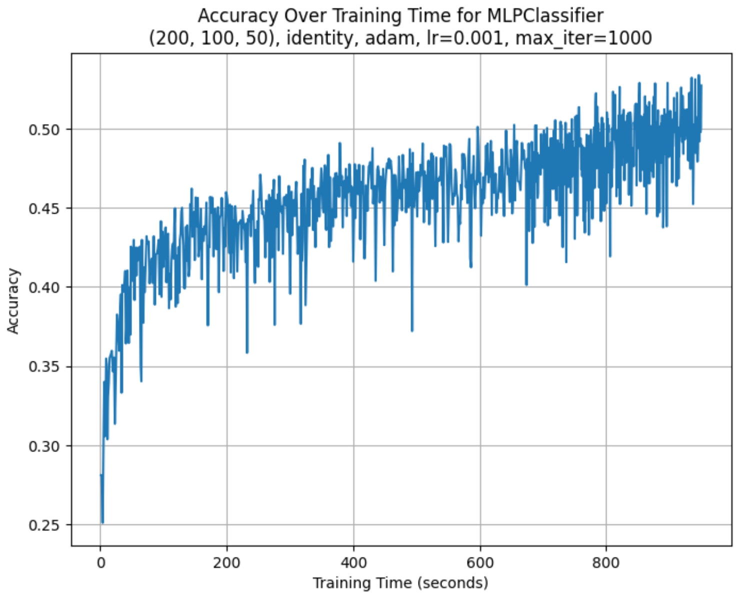
2. **Test 2:**

    - **Configuration:** Hidden Layer Sizes = (200, 200, 100, 100, 50, 50), Activation = 'ReLU', Solver = 'Adam', Learning Rate = 0.001, Max Iter = 1000
    - **Final Accuracy:** 0.992
    - **Total Training Time:** 196.264 seconds



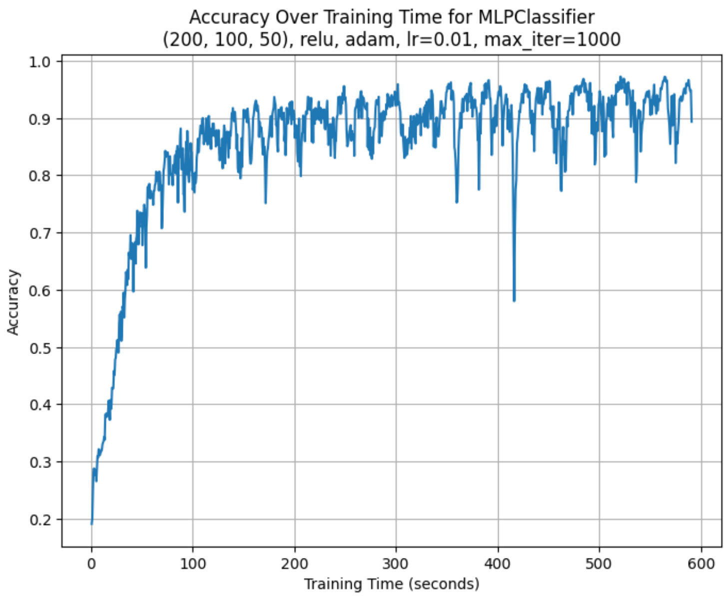
3. **Test 3:**

    - **Configuration:** Hidden Layer Sizes = (200, 100, 50), Activation = 'Identity', Solver = 'Adam', Learning Rate = 0.001, Max Iter = 1000
    - **Final Accuracy:** 0.528
    - **Total Training Time:** 950.269 seconds



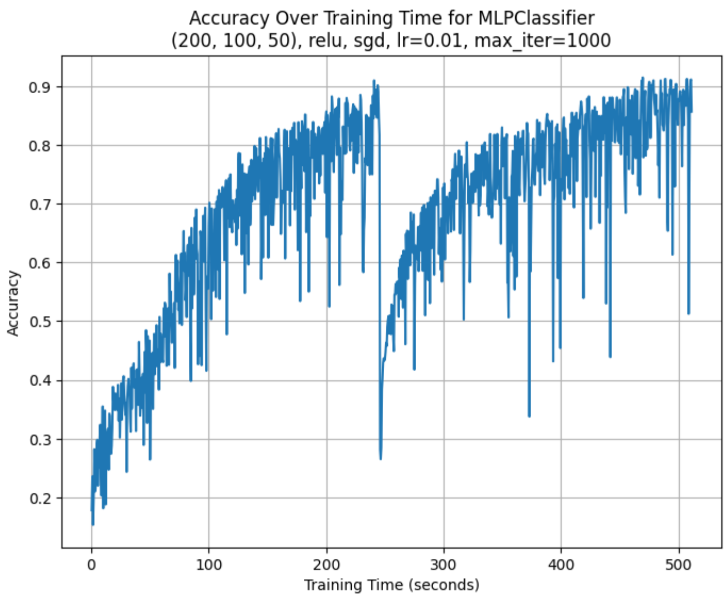
4. **Test 4:**

    - **Configuration:** Hidden Layer Sizes = (200, 100, 50), Activation = 'ReLU', Solver = 'Adam', Learning Rate = 0.01, Max Iter = 1000
    - **Final Accuracy:** 0.894
    - **Total Training Time:** 590.874 seconds



5. **Test 5:**

    - **Configuration:** Hidden Layer Sizes = (200, 100, 50), Activation = 'ReLU', Solver = 'SGD', Learning Rate = 0.001, Max Iter = 1000
    - **Final Accuracy:** 0.857
    - **Total Training Time:** 511.058 seconds



Analysis:

- **Iteration Number:** Tests 1 and 2 exhibit a higher number of iterations, indicating smoother convergence, while Tests 3, 4, and 5 show variations.
- **Graph Fluctuation:** Tests 1 and 4 demonstrate smoother convergence with felr oscillations, whereas Tests 3 and 5 show more fluctuations, suggesting instability during training.
- **Final Accuracy:** Tests 2 and 1 achieve the highest final accuracies, folloId by Test 4. Tests 3 and 5 yield comparatively lolr accuracies.
- **Total Training Time:** Test 1 has the shortest training time, folloId by Test 5. Test 3 requires the longest training time, indicating inefficiency.

Conclusion:

- Configurations with ReLU activation and Adam optimizer tend to perform better in terms of accuracy and convergence speed.
- Smaller learning rates generally result in smoother convergence and higher final accuracy.
- The choice of activation function significantly impacts model performance, with non-linear functions like ReLU outperforming linear functions like Identity.
- Deeper network structures may yield higher accuracies but require longer training times.
- SGD optimizer, while viable, may require more careful tuning and longer training times compared to Adam.

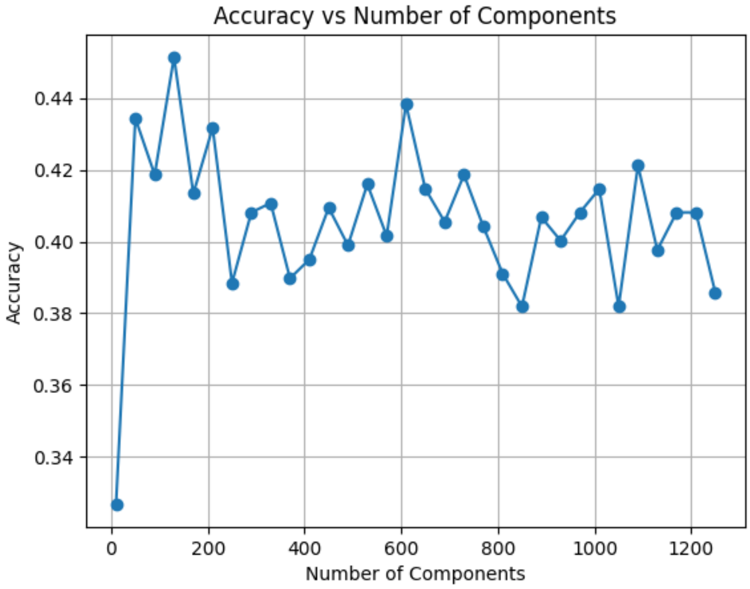
B) Report on the Role of Principal Component Analysis (PCA) in Classification Performance

Introduction: In this report, I investigate the impact of applying Principal Component Analysis (PCA) on the classification performance of a Multi-Layer Perceptron (MLP) classifier. The objective is to evaluate how the accuracy changes with different dimensions and gain insights into the original dataset's characteristics.

Methodology: I applied PCA to the dataset while varying the number of dimensions retained. The range of components tested ranged from 10 to 1250, with increments of 40. I utilized an MLPClassifier with predefined hyperparameters and evaluated the classification accuracy on the reduced datasets.

Results:

- The accuracy fluctuates as the number of components increases, generally following an upward trend until reaching a peak around 130 components.
- The highest accuracy achieved is approximately 45.1% when retaining 130 components.
- After around 650 components, the accuracy starts to decline, indicating that retaining too many components may not improve performance and could introduce noise or redundant information.



Interpretation:

1. **PCA's Role in Dimensionality Reduction:**
   - PCA aims to reduce the dimensionality of the dataset by identifying the most important principal components that capture most of the variance in the data.
   - By retaining only the most relevant components, PCA helps remove noise, reduce overfitting, and improve computational efficiency.
2. **Accuracy Fluctuation with Components:**
   - The fluctuations in accuracy as the number of components varies reflect a trade-off retaining enough information to capture underlying patterns and discarding irrelevant or redundant features.
   - The optimal number of components to retain may vary depending on the dataset and the specific problem. In this case, around 130 components seem to provide the best classification performance.

Conclusion:

- The original dataset likely contains a significant amount of redundant or irrelevant features.
- PCA aids in identifying a subset of components that capture the most important information while reducing dimensionality.
- Retaining too few components may lead to information loss, while retaining too many may introduce noise or increase complexity.
- The optimal number of components should be carefully chosen to balance classification accuracy, computational efficiency, and noise reduction.