**Question 1:**

It was necessary to develop better data visualization skills through Matplotlib practice in order to produce understandable and visually pleasing digit visualizations. In-depth PCA study, experimentation, and visualization helped me to overcome the challenges of understanding PCA ideas, calculating component counts, and evaluating findings. Tackled the unfamiliar Incremental PCA with documentation review and experimentation. Data transformation errors were minimized by careful testing and function selection. Assigning distinct colors in scatter plots for label-based coding was resolved through color map exploration. Interpreting PCA outcomes improved with dedicated study and tutorial consultation. Choosing principal components involved experimentation and informative tools like scree plots. And understanding the workflow from the dataset loading to result display was achieved through systematic breakdown and reference to documentation and tutorials.

**Question 2:**

The code had robust data analysis pipeline where Kernel Principal Component Analysis (kPCA) and perform hyperparameter tuning using GridSearchCV were explored. Then I generated a synthetic Swiss roll dataset, which was visually present in a 3D scatter plot. Next, applied kPCA with different kernels, namely linear, RBF, and polynomial. Each kernel introduced unique hyperparameter challenges, and I adjusted some parameters like gamma and degree. To effectively compare the results, I created subplots that visualize the transformed data points. I then employ GridSearchCV to optimize the hyperparameters of a Logistic Regression classifier, considering various kernel options and gamma values for kPCA. Finally, GridSearchCV results were scanned properly, providing a clear comparison of mean test scores across different gamma values and kernels. In conclusion, this code offers a comprehensive framework for dimensionality reduction and hyperparameter optimization.