Problem 3 – Classification (40%)

References for code:

- For learning how to plot k-means clustering, clusters by color, and etc.
- For fixing errors regarding column vs. 1d arrays.
- For creation of classification reports.
- For creation of boxplots.
- For reference to utilize ANOVA.

Note: Near the end of the report, there are some observations regarding the data that are unrelated to the prompt, but still important as it explains some choices made during the assignment.

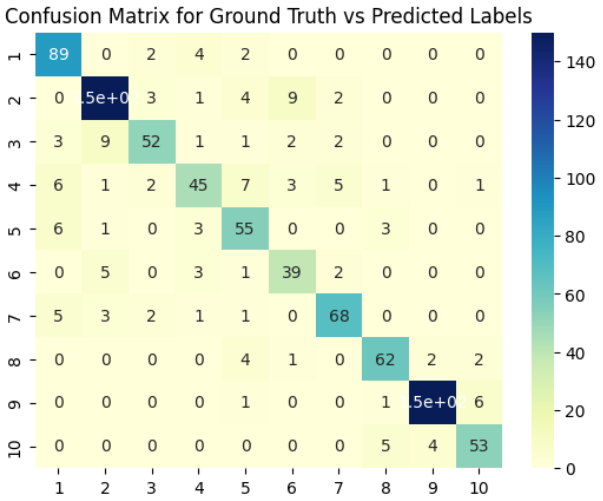
10% credit. Use a 1-nearest neighbor (i.e., k=1) classifier to predict labels for the validation samples (x2.csv). Report classification performance by comparing ground truth labels (c2.csv) against predictions from the 1NN classifier. Generate a confusion matrix and interpret the results.

When creating a 1-nearest neighbor classification model, I first used dimensionality reduction on the data using Fisher's LDA with 9 components. 6 components were used, as in problem 2, the scree plot showed that using 6 components allowed for capture of 95% of the variance in the data, and 9 components were required to capture 100% of the data. Thus, after using the training data to obtain the LDA eigenvector matrix, this matrix was used to project the training, validation, and test samples.

After this, the projected training data was used for our KNN Classifier, which was thus used to predict the labels for the projected validation data set.

The report of classification performance is shown below after comparing the ground truth labels against the predictions. This table will give us more descriptive information that the confusion matrix cannot show. The Confusion Matrix generated for comparison between the ground truth values and the predicted values are as follows:

	precision	recall	f1-score	support
Baked Products	0.82	0.92	0.86	97
Vegetables and Vegetable Products	0.89	0.89	0.89	169
Soups, Sauces, and Gravies	0.85	0.74	0.79	70
Sweets	0.78	0.63	0.70	71
Fast Foods	0.72	0.81	0.76	68
Fruits and Fruit Juices	0.72	0.78	0.75	50
Breakfast Cereals	0.86	0.85	0.86	80
Poultry Products	0.86	0.87	0.87	71
Beef Products	0.96	0.95	0.96	158
Lamb, Veal, and Game Products	0.85	0.85	0.85	62
accuracy			0.85	896
macro avg	0.83	0.83	0.83	896
weighted avg	0.85	0.85	0.85	896



Interpret the results.

- Based on the results above, we can conclude that the 1-nearest neighbor classifier is relatively successful when it comes to classification accuracy, as it had a classification accuracy of 85.2% (which was calculated by dividing the number of correctly classified samples over total examples. The 1-nearest neighbor model especially was successful for properly classifying classes 9 and 2, or Beef Products and Vegetable/Vegetable Products. What was especially interesting was that the model was able to predict if a sample was a Beef Product with 96% accuracy, as noted by the precision column.
- The model struggled the most with predicting whether or not a sample was a Fast Food or Fruit/Fruit Juice, as the model was only able to correctly predict these classes 72% of the time. Further inspection into the classification matrix shows that Fast Foods, or Class 5 samples, were often misclassified as class 4 or sweets. This aligns with our current understanding of this food group, as the nutritional profiles of Fast Food and Sweets are similar (some examples of features that may be similar could include Total Sugars, Total Lipids, etc. but would require further observation of the LDs and how the LDs are weighted).
- Class 6, or Fruit/Fruit Juice was often misclassified as Vegetables and Vegetable Products, which aligns with an observation found in p2. In terms of organization into larger food groups, Fruit/Fruit Juice and Vegetable/Vegetable products were close to each other, which aligns with our current understanding of their nutritional profiles.
- Some quick misclassifications are noted below:
    - Class 1 was often misclassified as 4, 5, and 7
        - Baked Products misclassified as Sweets, Fast Foods, and Breakfast Cereals.

- Class 2 was often misclassified as 3
    - Vegetable/Vegetable Products vs. Soups, Sauces, and Gravies
- Class 10 was often misclassified as 9
    - This also aligns with our previous observations in terms of similarity in the nutritional profiles of Beef Products vs. Lamb, Veal, and Game Products.

20% credit. Perform cross-validation to find the optimum number of neighbors (k) for the kNN classifier. For this purpose, merge the training and validation sets ([x1; x2], [c1; c2]), and then re-split them following the original proportions. For each split, use the training samples to classify the validation samples, and store the classification rate. Repeat the re-splitting process multiple times and compute the average performance. Start with k=1 neighbors, and repeat for k=2, 3, …

When performing cross-validation to find the optimum number of neighbors for the kNN classifier, I first merged the x1.csv and x2.csv dataset to create a set containing all the features, and merged the y1.csv and the y2.csv dataset to create a set containing all the classes. This gave us a new dataset with the following dimensions:

For the new merged dataset of features, we had a dataset containing 1794 rows and 46 features and a dataset of classes containing the same number of rows but one column. Then, similarly to part 1, LDA was applied to the merged dataset using 9 components. The data was thus projected and underwent dimensionality reduction, which resulted in a dataset with the dimensions 1794 rows by 9 columns or features.

I tested with 1-15 neighbors, with each neighbor model being tested 25 times with the resplitting process.

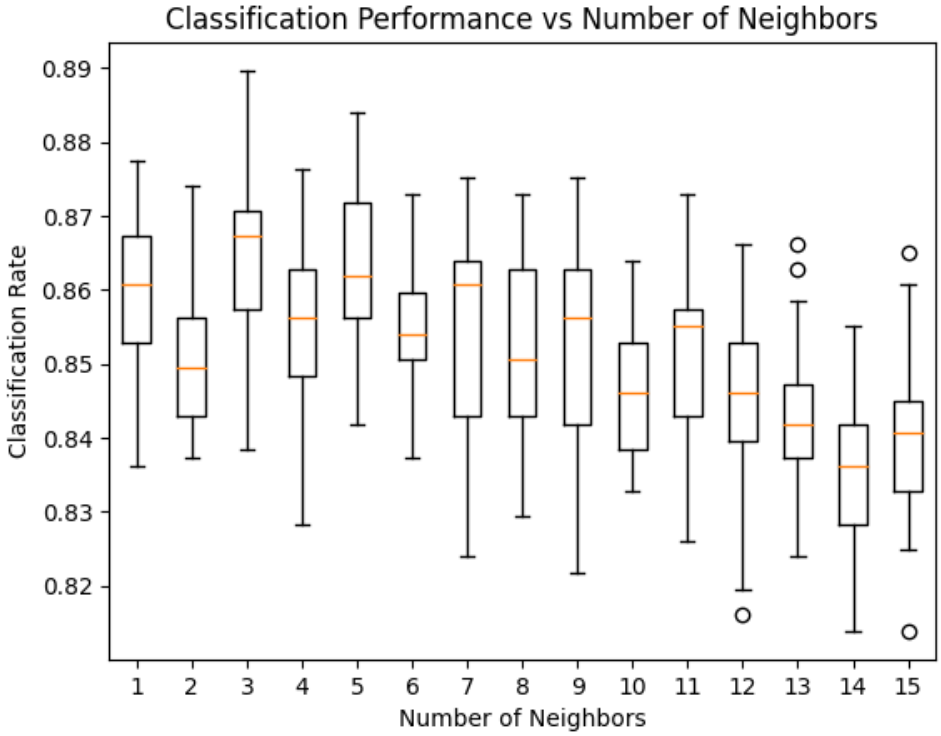
Plotting the average classification performance vs. Number of Neighbors gives us the following, with the optimum number of neighbors being 3. For this particular run, the average performance of the model that uses k = 3 neighbors has a classification accuracy of 86.5%.

Interpret the results.

- *Discuss your findings.Does classification performance improve as you increase the number of neighbors?*
    - Based on the graph above, it appears that there is a trend of decreased classification performance as the number of neighbors increases. This aligns with what we know about kNN in regards to the choosing k. When k grows to be too large, it destroys the locality of the class prediction. This is because as k increases, more examples are taken into account.

- One example of this as an extreme is if we set the number of neighbors to be equal to the number of samples in the training set. Thus, we identify each new point to be the largest occurrence of a class in the training set, and thus all new points are predicted to be the class.

Creating a box plot for each value of k evaluated validates this observation.

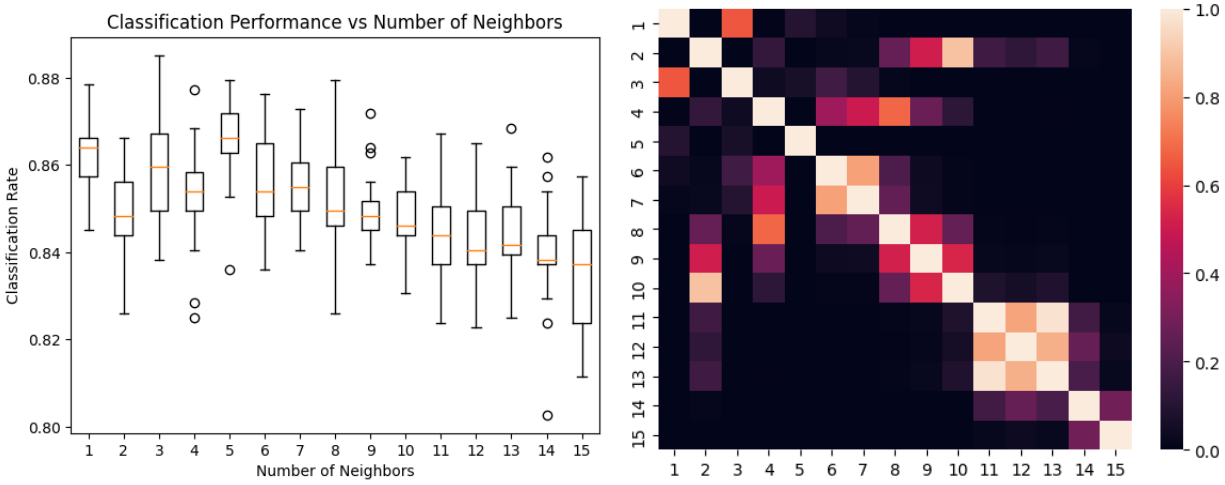


Interpret the results.

Are changes in classification performance statistically significant (i.e., as indicated by ANOVA)? Generate a box plot for each value of k you evaluated.

- In order to determine this we must perform ANOVA in order to see if the classification performance is statistically significant.
- Performing an ANOVA test determines if there is statistically significant difference between the groups. Our null hypothesis is that there is no significant difference between the groups, and that all group means are equal. Our alternative hypothesis is that there is a significant difference between at least one pair of the groups in the set of groups.
- We choose a significance level of 0.05, and get a p value of 3.4413358511405296e-24, which tells us that there is a significant difference between the classification performances of the k values.
- We thus have enough evidence to conclude that there is significant difference between at least one pair of groups.

When comparing the means of different groups (we use a different classification performance boxplot as I had to regenerate the data), we get the following heatmap of p values:



This shows us that there does exist significant differences between different groups, as some groups, such as k = 1 and k = 15 have statistically significant differences in classification performance. For the sake of this, our null hypothesis was also that there is no significant difference between groups k = 1 and k = 15, and because the p value was less than 0.05, we reject the null hypothesis and say that there is a significant difference in the classification rate means of models that use 1-nearest neighbor and 15-nearest neighbors.

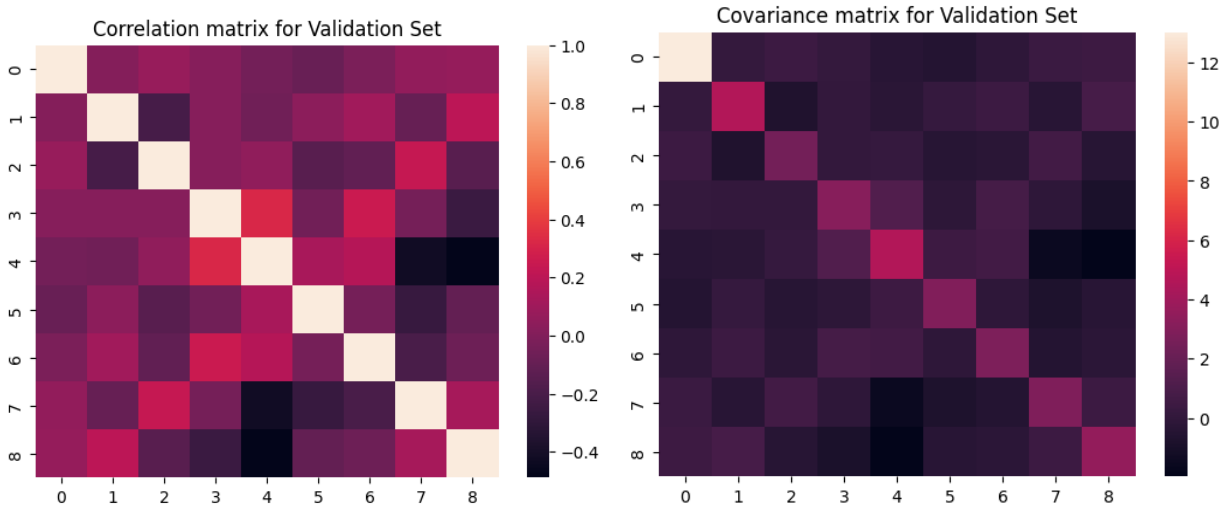
Box plot interpretation:

- In terms of the boxplots above (for the sake of this, we reference the first boxplot shown)

10% credit. Repeat part a. using a different classifier from those discussed in lecture 8, e.g., nearest mean classifier, quadratic classifier . You may incorporate prior probabilities to improve performance.

For the sake of this part, I decided to use nearest mean classifier using the original train and validation sets. Because I already took the correlation and covariance matrix for the testing set, which is shown below in the index, I thus take the correlation and covariance matrix of the validation set following projection of the features using the testing set to create the eigenvector

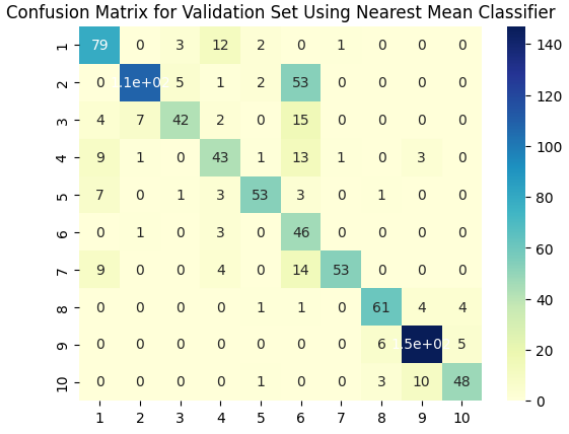
matrix used for projection. This gives me the following correlation and covariance matrices:



Based on the above, I will not use the diagonal matrix and assume that features are uncorrelated. This is a very strong assumption, and in the case of the validation set, there appears to be moderate to strong correlation between certain features based on the correlation matrix for the validation set, even after dimensionality reduction.

Following fitting of the model using the projected training values and prediction of the projected validation set, we get the following classification report and confusion matrix:

	precision	recall	f1-score	support
Baked Products	0.73	0.81	0.77	97
Vegetables and Vegetable Products	0.92	0.64	0.76	169
Soups, Sauces, and Gravies	0.82	0.60	0.69	70
Sweets	0.63	0.61	0.62	71
Fast Foods	0.88	0.78	0.83	68
Fruits and Fruit Juices	0.32	0.92	0.47	50
Breakfast Cereals	0.96	0.66	0.79	80
Poultry Products	0.86	0.86	0.86	71
Beef Products	0.90	0.93	0.91	158
Lamb, Veal, and Game Products	0.84	0.77	0.81	62
accuracy			0.76	896
macro avg	0.79	0.76	0.75	896
weighted avg	0.82	0.76	0.77	896



Classification Accuracy for the Nearest Mean Classifier Model was 0.7589285714285714.

Interpret the results.

NOTE: depending on the data split, the covariance matrix of each class may be ill-conditioned, so you may need to use a diagonal matrix (i.e., which assumes features are uncorrelated). Is this assumption valid?

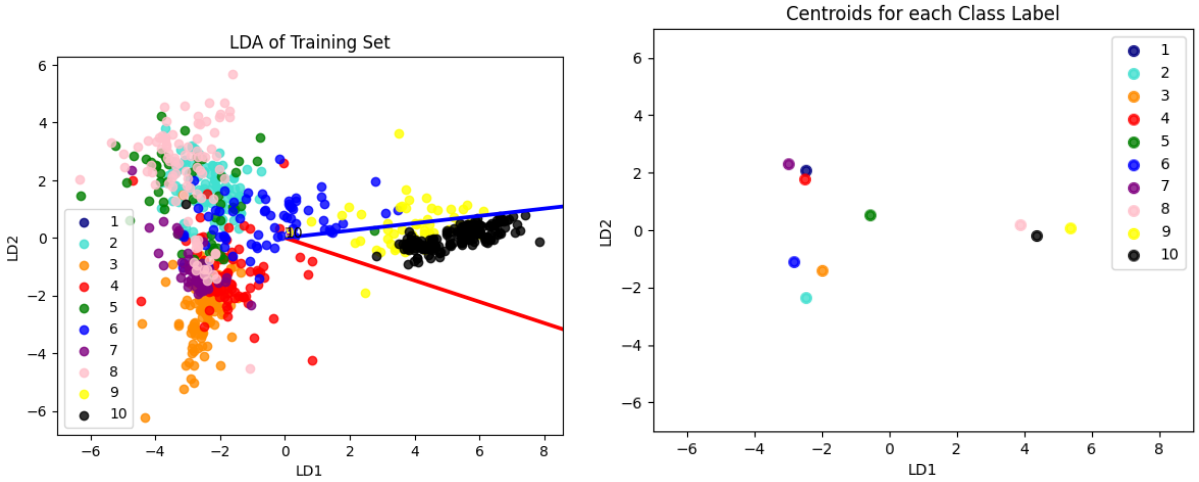
- When plotting the correlation and covariance matrices of the validation and training data following LDA transformation, I noticed that the correlation of the

training set was uncorrelated following transformation while the validation set appeared to have some correlation between features.

- These results match expectations, as we used the training set to create the
- In the case of the assumption that features are uncorrelated, creating a correlation matrix of the unprocessed features of the training set shows that there are moderate to strong positive correlations between different features. Though the correlations show that there is a relationship between the two features, it cannot be said with certainty that one feature causes another or that these correlations are not the result of random sampling error. Even so, though we are not working with the raw data for the nearest mean classification model, there is reason to believe that there are correlations between the dimensionally reduced features, thus, we do not assume that the features are uncorrelated.

Discussion of results

- In terms of the nearest mean classifier, the model had an accuracy rate of 75.9%, which was lower than the accuracy of both the kNN model using 1-nearest neighbor and the best performing kNN model for this run, which used 3 neighbors and had a classification success rate of 88%.
- Because nearest mean classification is processes the data beforehand to generate a centroid for each class, looking back to the distribution of points in the LDA 2d scatterplot aligns with the results of the accuracy of the nearest means classifier model:



For the sake of nearest mean classification, we used the original training set. Looking at the 2d scatter plot of the means, we see that the means of each class is not a good representation of whether or not a sample is part of a class or not, as for nearest-mean classifier to be optimized, the covariance matrices for all classes need to be equal (at least if we use mahalanobis distance) . Based on the above spread and variance for the 2d scatter plot projection, if we use euclidean distance classification for the nearest mean classifier, we will get unideal results, as the means of some classes are located near each other. This makes decision boundaries that incorrectly classify some classes, as based on the nearest means classifier created, many points from the training set would also be misclassified. For example, in the case of the training

set, many points classified as class 10 would be incorrectly classified as class 8 or 9 due to the locality of the class means.
Interestingly enough, though, the nearest mean classifier outperformed the 1-nearest neighbor model depending on the class. Though nearest neighbor performed inefficiently for Fruits/Fruit Juices, as it correctly predicted the class 32% of the time, The nearest means classifier had better performance for classifying vegetables, Fast Foods, and Breakfast Cereals (which had the maximum accuracy of this model as the nearest means classified it correctly 96% of the time).
Looking at the confusion matrix tells that for Fruits/Fruit Juices samples, the model was more likely to classify the sample as a Vegetable/Vegetable product 36% of the time. Looking at the centroids above aligns with this result, as the mean for class 6 and class 2 are located close to each other, meaning that points classified as class 6 are likely to also be misclassified as other classes in that general locality, including class 3 and 4.

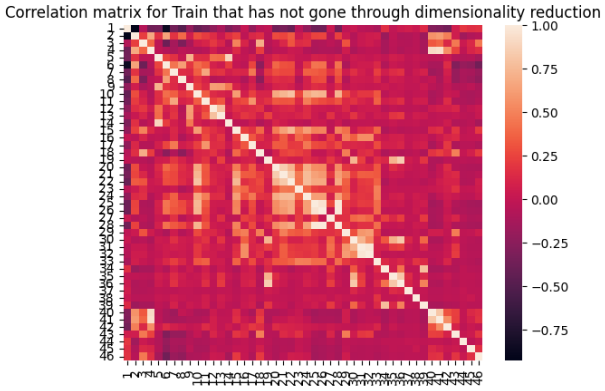
5% extra credit. Write a Jupyter notebook that reads a test dataset (x3.csv) and generates predictions (c3_pred.csv). In this case, you will not be able to evaluate the performance of this classifier since the test labels (c3.csv) are not available to you. Instead, we will compare your predictions against the correct class labels (which we have kept aside). NOTE: to receive extra credit, we will need to run your code and generate the predictions file (c3_pred.csv) ourselves. In other words, you will not receive extra credits if you only submit the file c3_pred.csv.

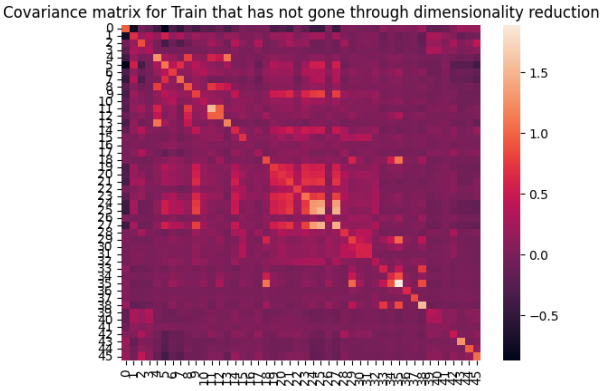
Interpret the results.

- There are no results to be interpreted, as this is the prediction generator. For the sake of this problem, I use the best performing model from my K-Fold cross validation model by first determining the optimum neighbors model, and out of those models with k neighbors, I choose the best performance model with that number of neighbors.

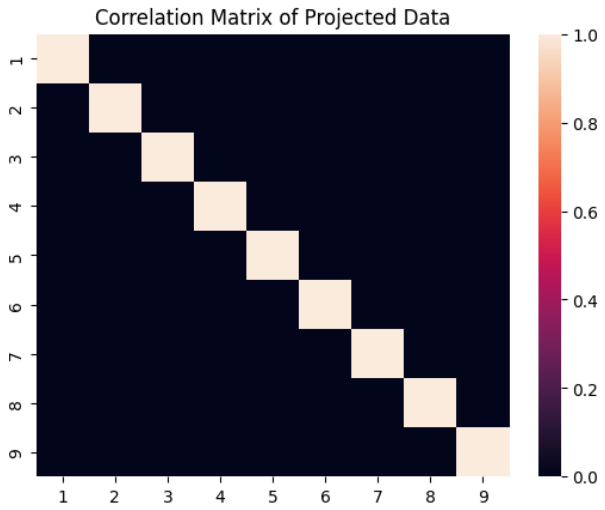
INDEX: NOT RELATED TO THE PROBLEM PROMPTS, but still important Initial Data Exploration:

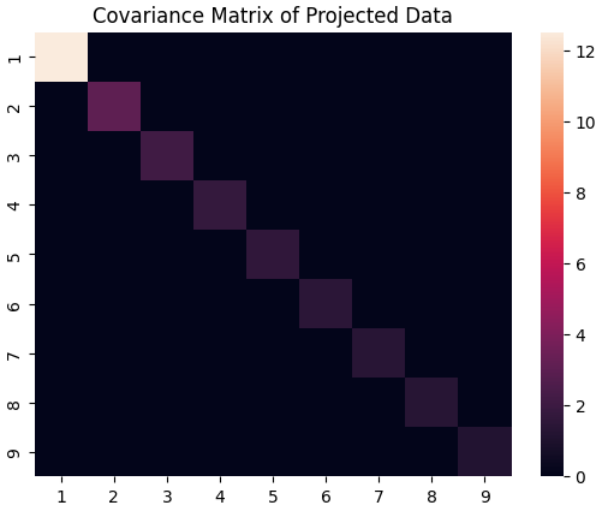
Before proceeding with the assignment, I first examined the covariance matrix for the 46 features to see if there were any features that were collinear, or contained high correlation with each other. Because we are dealing with 46 features, we are faced with the curse of dimensionality, and I wanted to see what the covariance and covariance matrix would look like after applying LDA to the data, as we will be using the dimensionality reduction to project the data into a lower dimension.
Doing so gives us the following for all 46 features:



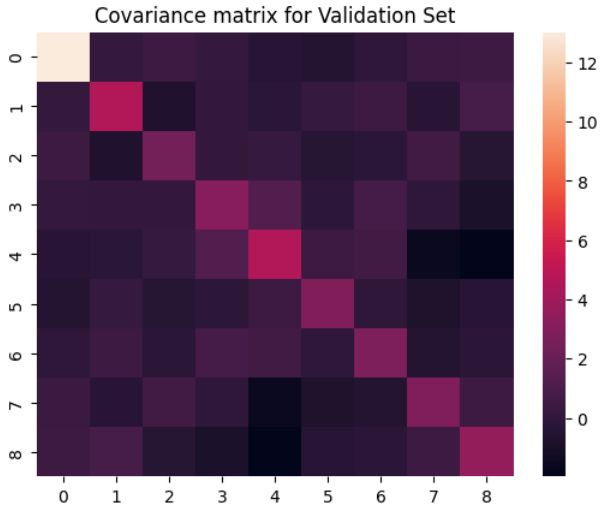


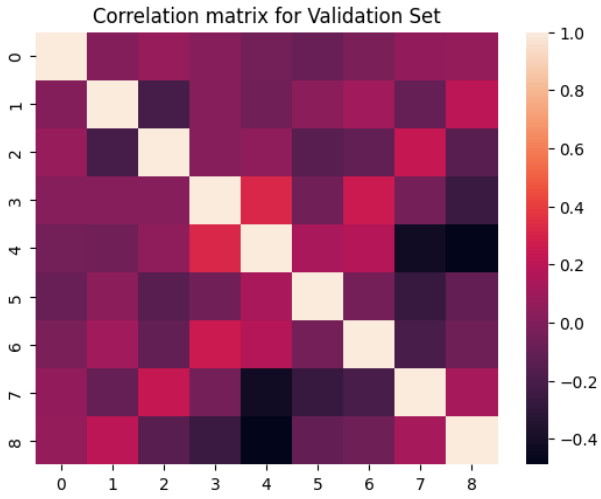
Because in the previous problem p2, we determined that 6 components were needed to capture 95% of the variance in the data and 9 components were needed to capture 100% of the variance, I thus performed LDA on the training data using 9 components. Following projection of the features, we get the following:





Though it appears that there is no correlation between the features for the components following application of LDA, when we look at the covariance and correlation matrices of the validation set, we do see that there exists some moderate to high correlation between the features. For this reason, we choose not to use the diagonal matrix. (This is also mentioned above)


Covariance matrix for Validation Set
0
1
2
3
4
5
6
7
8
12
10
8
6
4
2
0


Correlation matrix for Validation Set
0
1
2
3
4
5
6
7
8
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4