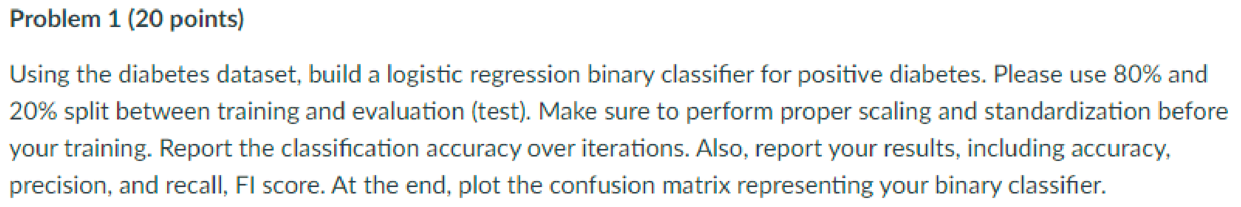
HW3.ipynb

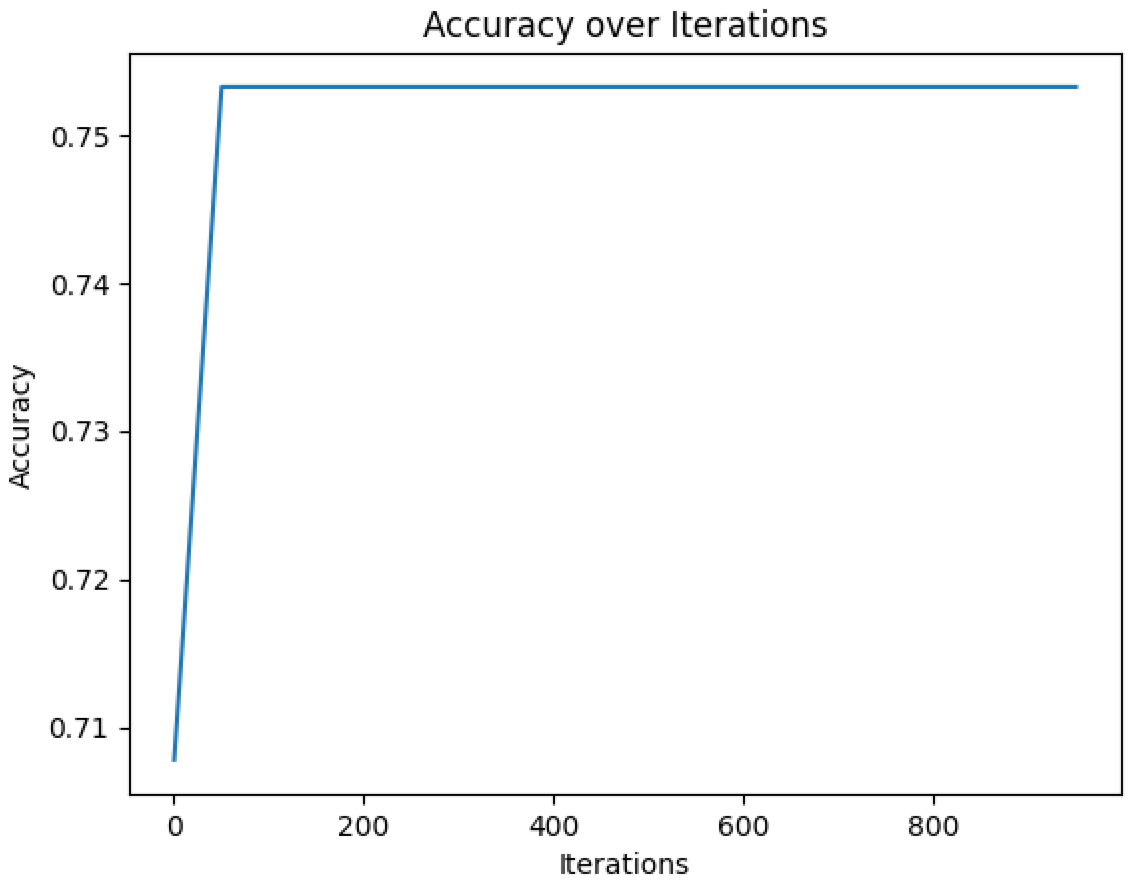
https://colab.research.google.com/drive/1L0sW-94KwHhCFwfeKdTVMz-nglbsl_hP?authuser=3#scrollTo=hh4FMNQQUB80

https://github.com/YousefALH/MLHWCore/tree/main/Homework/Homework%203

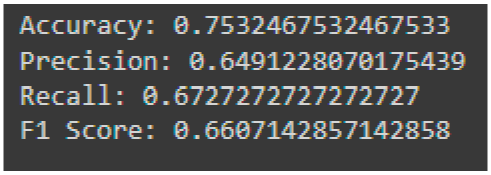


Problem 1 (20 points)

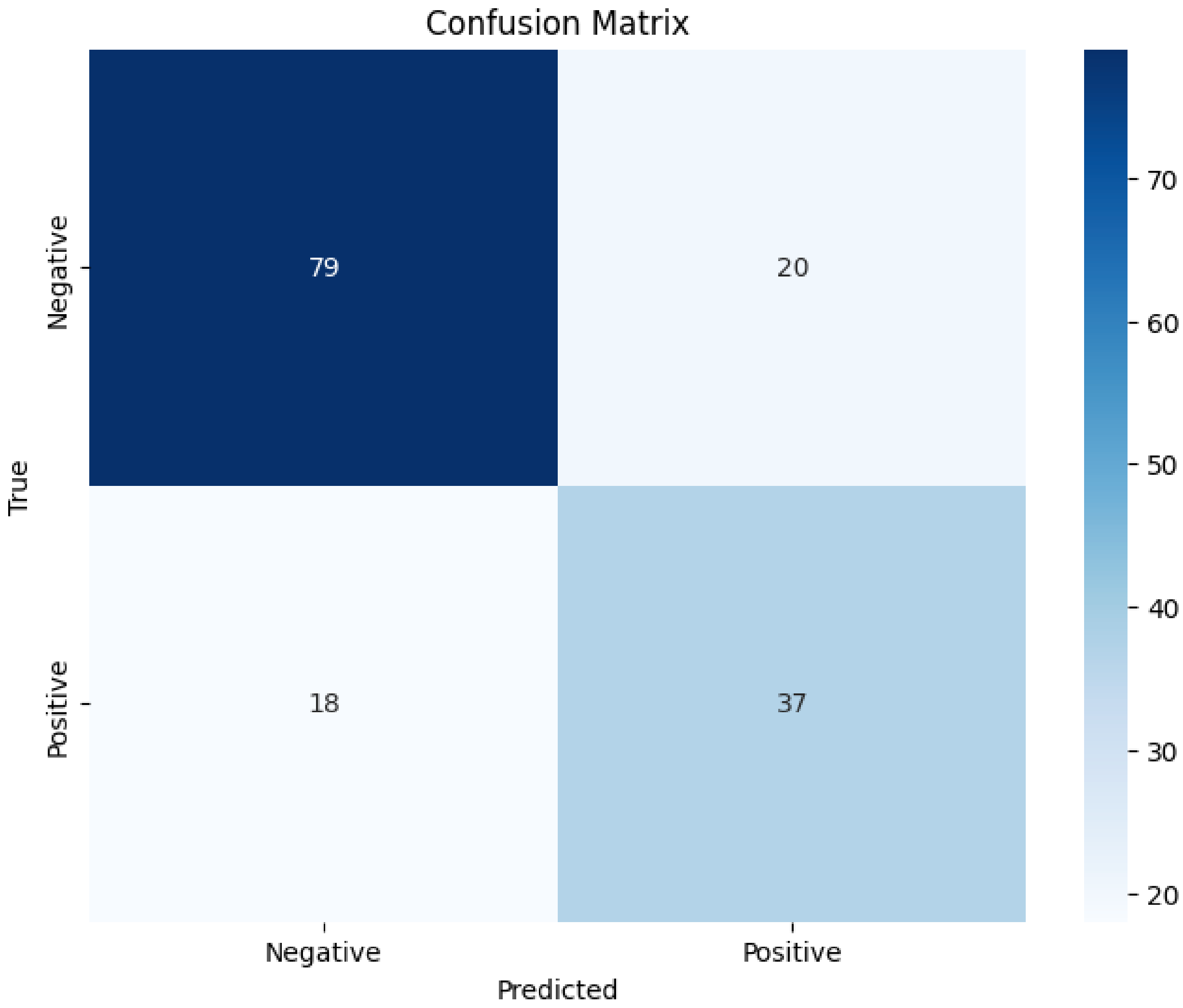
Using the diabetes dataset, build a logistic regression binary classifier for positive diabetes. Please use 80% and 20% split between training and evaluation (test). Make sure to perform proper scaling and standardization before your training. Report the classification accuracy over iterations. Also, report your results, including accuracy, precision, and recall, FI score. At the end, plot the confusion matrix representing your binary classifier.



Iteration is at 1000.



```
Accuracy: 0.7532467532467533
Precision: 0.6491228070175439
Recall: 0.6727272727272727
F1 Score: 0.6607142857142858
```

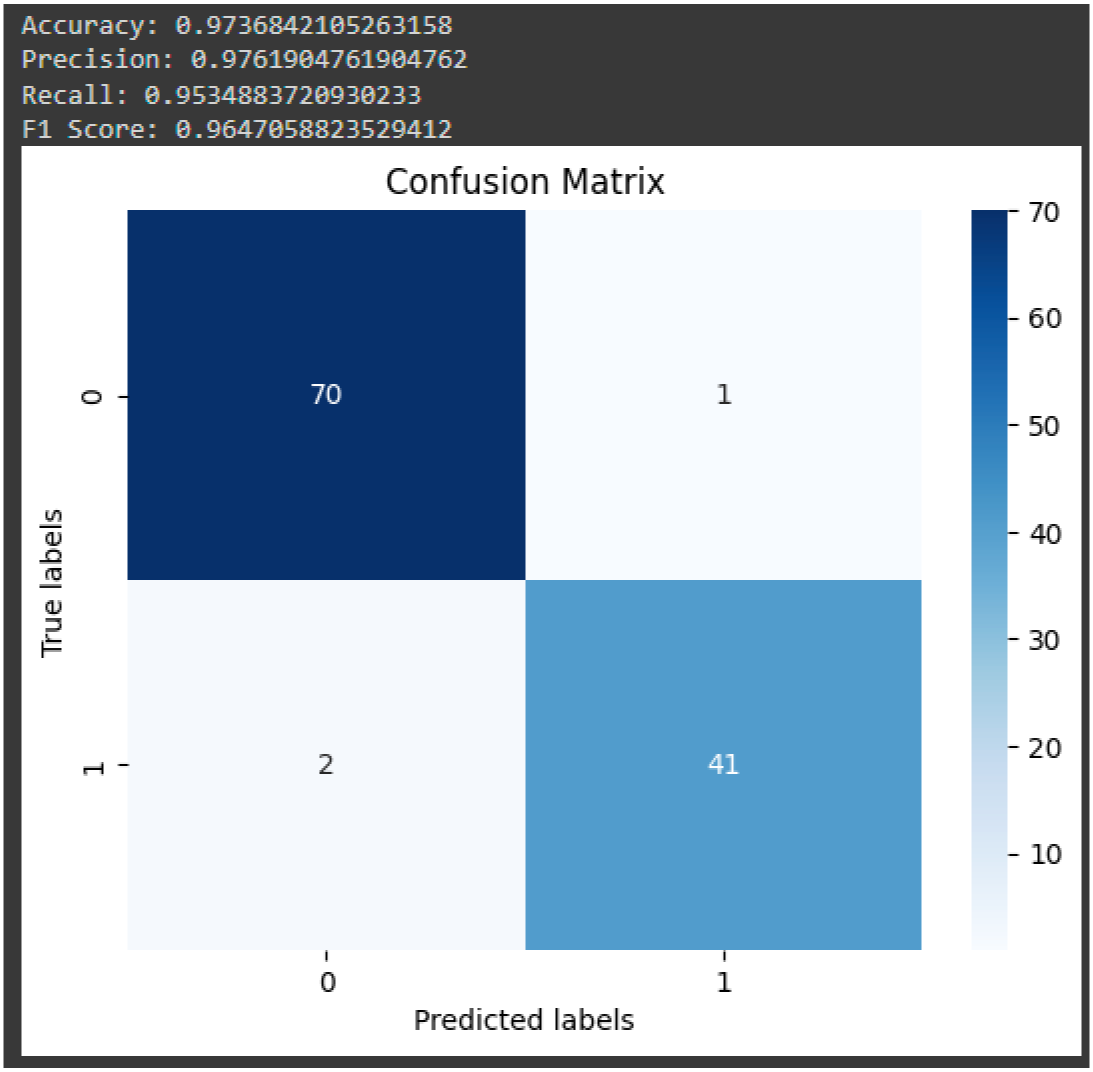


Problem 2 (20pts):

a. Use the cancer dataset to build a logistic regression model to classify the type of cancer (Malignant vs. benign). First, create a logistic regression that takes all 30 input features for classification. Please use 80% and 20% split between training and evaluation (test). Make sure to perform proper scaling and standardization before your training. Report your classification accuracy over iterations. Also, report your results, including accuracy, precision, recall, and F1 score. At the end, plot the confusion matrix representing your binary classifier.

b. How about adding a weight penalty here, considering the number of parameters? Add the weight penalty repeat the training, and report the results.

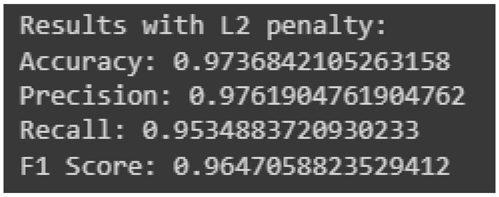
a)



```
Accuracy: 0.9736842105263158
Precision: 0.9761904761904762
Recall: 0.9534883720930233
F1 Score: 0.9647058823529412
```

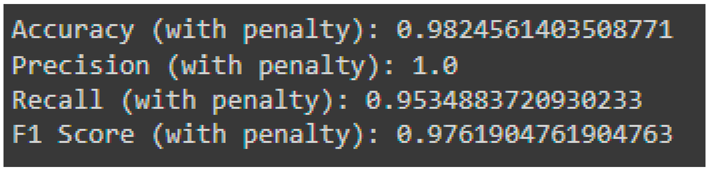
2B

Approach 1 L2 penalty is from the sklearn.linear_model._logistic.LogisticRegression instance

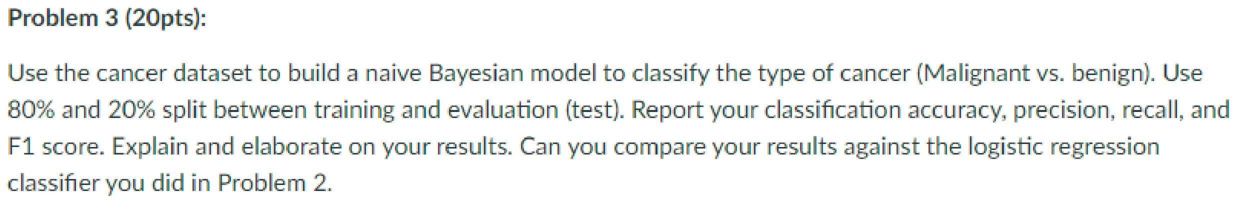


```
Results with L2 penalty:
Accuracy: 0.9736842105263158
Precision: 0.9761904761904762
Recall: 0.9534883720930233
F1 Score: 0.9647058823529412
```

Approach 2 for this one is more controlled “model_regularized = LogisticRegression(max_iter=1000, C=0.1)”

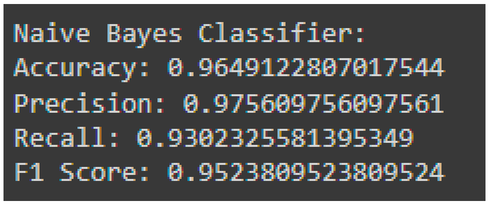


```
Accuracy (with penalty): 0.9824561403508771
Precision (with penalty): 1.0
Recall (with penalty): 0.9534883720930233
F1 Score (with penalty): 0.9761904761904763
```

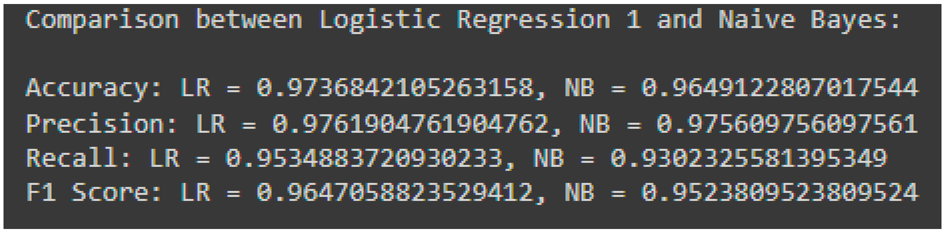


Problem 3 (20pts):

Use the cancer dataset to build a naive Bayesian model to classify the type of cancer (Malignant vs. benign). Use 80% and 20% split between training and evaluation (test). Report your classification accuracy, precision, recall, and F1 score. Explain and elaborate on your results. Can you compare your results against the logistic regression classifier you did in Problem 2.

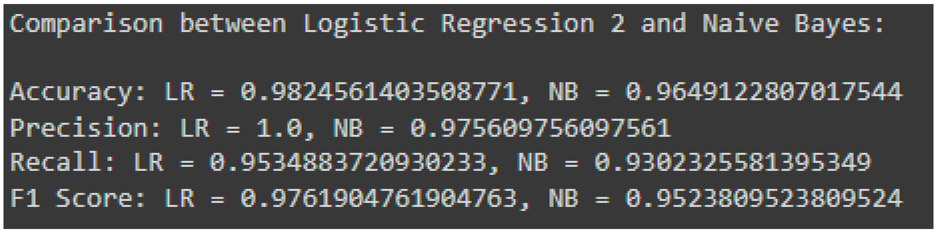


```
Naive Bayes Classifier:
Accuracy: 0.9649122807017544
Precision: 0.975609756097561
Recall: 0.9302325581395349
F1 Score: 0.9523809523809524
```



```
 Comparison between Logistic Regression 1 and Naive Bayes:

 Accuracy: LR = 0.9736842105263158, NB = 0.9649122807017544
 Precision: LR = 0.9761904761904762, NB = 0.975609756097561
 Recall: LR = 0.9534883720930233, NB = 0.9302325581395349
 F1 Score: LR = 0.9647058823529412, NB = 0.9523809523809524
```

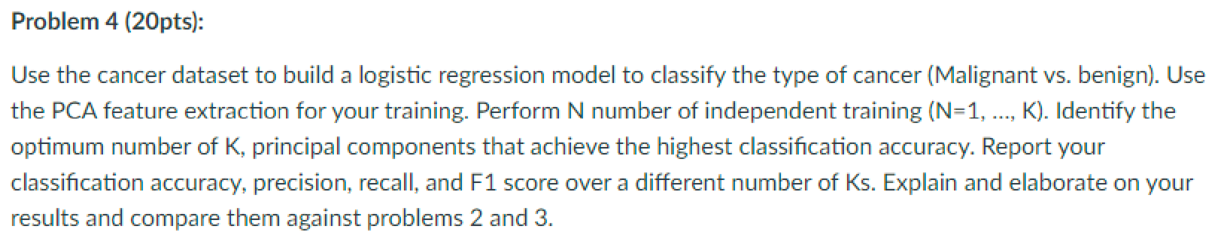


```
Comparison between Logistic Regression 2 and Naive Bayes:

Accuracy: LR = 0.9824561403508771, NB = 0.9649122807017544
Precision: LR = 1.0, NB = 0.975609756097561
Recall: LR = 0.9534883720930233, NB = 0.9302325581395349
F1 Score: LR = 0.9761904761904763, NB = 0.9523809523809524
```

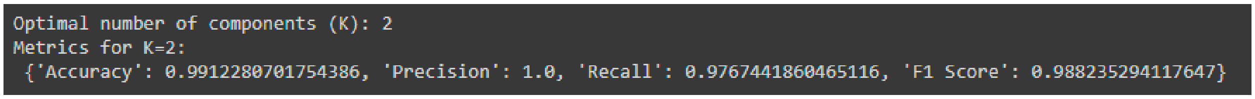
Comparison against Logistic Regression:
The logistic regression classifier and the Naive Bayes classifier approach the classification problem from different angles:

Model Complexity: Logistic Regression is more flexible and can capture more complex relationships in the data because it tries to estimate parameters to best separate the classes. Naive Bayes assumes all features are independent given the class label, which is often not true in real-world scenarios (hence "naive"). This makes Naive Bayes often less accurate.
Speed: Naive Bayes can be faster because it only calculates probabilities using feature statistics.
Interpretability: Both models provide different kinds of interpretability. Logistic regression provides coefficients which can give insights into feature importance, while Naive Bayes gives probabilities which can be easier to understand.

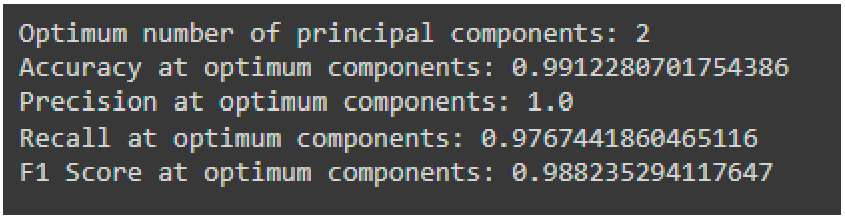


Problem 4 (20pts):

Use the cancer dataset to build a logistic regression model to classify the type of cancer (Malignant vs. benign). Use the PCA feature extraction for your training. Perform N number of independent training (N=1, …, K). Identify the optimum number of K, principal components that achieve the highest classification accuracy. Report your classification accuracy, precision, recall, and F1 score over a different number of Ks. Explain and elaborate on your results and compare them against problems 2 and 3.



```
Optimal number of components (K): 2
Metrics for K=2:
 {'Accuracy': 0.9912280701754386, 'Precision': 1.0, 'Recall': 0.9767441860465116, 'F1 Score': 0.988235294117647}
```



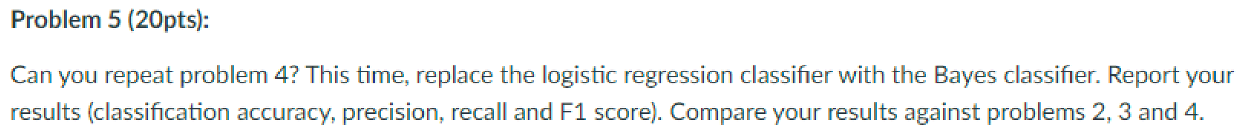
```
Optimum number of principal components: 2
Accuracy at optimum components: 0.9912280701754386
Precision at optimum components: 1.0
Recall at optimum components: 0.9767441860465116
F1 Score at optimum components: 0.988235294117647
```

Comparison against Problems 2 and 3:

Complexity: Introducing PCA adds an extra layer of complexity compared to the standard logistic regression in Problem 2 and the Naive Bayes classifier in Problem 3. But it can lead to faster training times since fewer features might be involved.

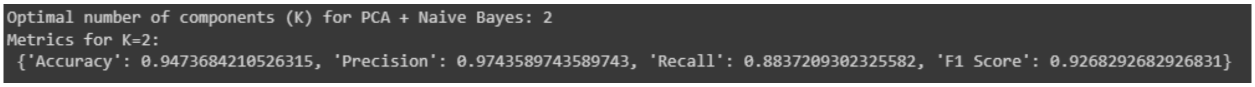
Interpretability: The introduction of PCA makes the model a bit harder to interpret. The principal components do not have a direct relationship with the original features, which can make it challenging to explain the results in a business context.

Performance: Depending on the dataset, PCA might increase or decrease performance. If there's a lot of redundant information in the features, PCA might improve the model. Conversely, if important variance is discarded, performance might decrease.



Problem 5 (20pts):

Can you repeat problem 4? This time, replace the logistic regression classifier with the Bayes classifier. Report your results (classification accuracy, precision, recall and F1 score). Compare your results against problems 2, 3 and 4.



```
Optimal number of components (K) for PCA + Naive Bayes: 2
Metrics for K=2:
 {'Accuracy': 0.9473684210526315, 'Precision': 0.9743589743589743, 'Recall': 0.8837209302325582, 'F1 Score': 0.9268292682926831}
```

Comparison:

PCA + Naive Bayes vs. Problem 2 (Logistic Regression):

Logistic Regression without PCA might capture more intricate patterns since it uses all the original features.

PCA + Naive Bayes will typically run faster due to reduced dimensionality but may miss some intricate patterns.

PCA + Naive Bayes vs. Problem 3 (Naive Bayes):

PCA can amplify or decrease the performance of Naive Bayes depending on how well the principal components represent the data.

Direct comparison of the metrics will highlight if PCA provides any benefit.

PCA + Naive Bayes vs. Problem 4 (PCA + Logistic Regression):

The comparison between PCA + Logistic Regression and PCA + Naive Bayes reflects the strengths and weaknesses of the base classifiers when used in conjunction with PCA.