Accuracy before deleting uninformative features

	Train Accuracy	Test Accuracy
Naïve bayes	0.81	0.78
KNN	0.87	0.78
Logistic Regression	0.93	0.93
Decision Tree	1.00	0.91
AdaBoost	0.94	0.93
SVM	0.72	0.68

	Train Accuracy	Test Accuracy
Custom Decision Tree	0.97	0.91
Custom KNN	0.82	0.76

Naïve bayes: the model performs well and the training and testing accuracies are close to each other, indicating a balance between training and testing.

KNN: It has good accuracy in training and low accuracy in testing, which indicates overfitting.

Logistic regression: The model has a very good balance between training and testing, so the model performs very well.

Decision tree: The model has fully learned the training data, but it has less experimental accuracy, overfitting.

AdaBoost: The model performs well.

SVM: has low accuracy in both cases, which indicates underfitting.

Run time before deleting uninformative features

	Train Time (sec)	Test Time (sec)
Naïve bayes	0.01	0.01
KNN	0.01	0.50

Logistic Regression	1.21	0.00
Decision Tree	0.09	0.00
AdaBoost	0.47	0.06
SVM	0.55	1.50

	Train Time	Test Time
Custom Decision tree	13.02	0.02
Custom KNN	0.00	5.59

Accuracy after using PCA

	Train Accuracy	Test Accuracy
Naïve bayes	0.70	0.66
KNN	0.87	0.77
Logistic Regression	0.89	0.90
Decision Tree	1.00	0.87
AdaBoost	0.93	0.91
SVM	0.72	0.68

	Train Accuracy	Test Accuracy
Custom Decision Tree	0.98	0.87
Custom KNN	0.82	0.77

Accuracy after using select K-best:

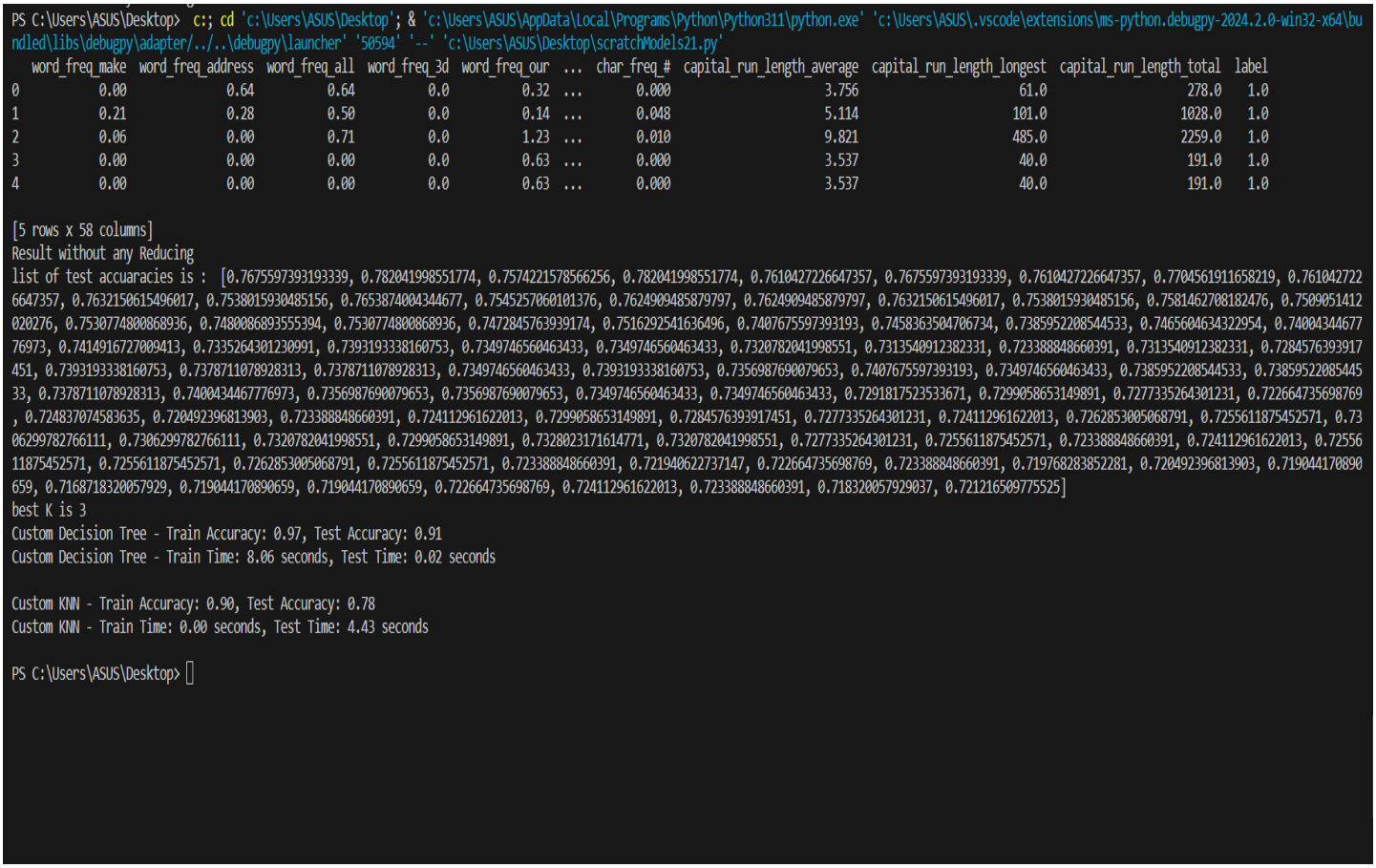
	Train Accuracy	Test Accuracy
Naïve bayes	0.87	0.87
KNN	0.87	0.78
Logistic Regression	0.89	0.88
Decision Tree	1.00	0.89
AdaBoost	0.93	0.92
SVM	0.70	0.67

	Train Accuracy	Test Accuracy
Custom Decision Tree	0.97	0.91
Custom KNN	0.80	0.75

Analysis: The accuracy has decreased after implementing dimensionality reduction techniques, this could mean that the features of our data are all useful and useful for training and testing, or it could be a sign that we have not been able to identify the features well. to remove uninformative data and this requires the implementation of more powerful techniques.

In the following, we found the best k for the Custom KNN algorithm, with a for loop, a list of accuracies was obtained for each state of K, and the highest value of that list of K was considered. The optimal K value was 3, which yielded an accuracy of 0.78 on the test data and an accuracy of 0.90 on the training data.

The accuracy values for K from 2 to 100 are as follows output from the code.



list of test accuracies is : [0.7675597393193339, 0.782041998551774, 0.7574221578566256, 0.782041998551774, 0.7610427226647357, 0.7675597393193339, 0.7610427226647357, 0.7704561911658219, 0.7610427226647357, 0.7632150615496017, 0.7538015930485156, 0.7653874004344677, 0.7545257060101376, 0.7624909485879797, 0.7624909485879797, 0.7632150615496017, 0.7538015930485156, 0.7581462708182476, 0.7509051412020276, 0.7530774800868936, 0.7480086893555394, 0.7530774800868936, 0.7472845763939174, 0.7516292541636496, 0.7407675597393193, 0.7458363504706734, 0.7385952208544533, 0.7465604634322954, 0.7400434467776973, 0.7414916727009413, 0.7335264301230991, 0.7393193338160753, 0.7349746560463433, 0.7349746560463433, 0.7320782041998551, 0.7313540912382331, 0.723388848660391, 0.7313540912382331, 0.7284576393917451, 0.7393193338160753, 0.7378711078928313, 0.7378711078928313, 0.7349746560463433,

0.7393193338160753, 0.7356987690079653, 0.7407675597393193, 0.7349746560463433, 0.7385952208544533, 0.7385952208544533, 0.7378711078928313, 0.7400434467776973, 0.7356987690079653, 0.7356987690079653, 0.7349746560463433, 0.7349746560463433, 0.7291817523533671, 0.7299058653149891, 0.7277335264301231, 0.722664735698769, 0.724837074583635, 0.720492396813903, 0.723388848660391, 0.724112961622013, 0.7299058653149891, 0.7284576393917451, 0.7277335264301231, 0.724112961622013, 0.7262853005068791, 0.7255611875452571, 0.7306299782766111, 0.7306299782766111, 0.7320782041998551, 0.7299058653149891, 0.7328023171614771, 0.7320782041998551, 0.7277335264301231, 0.7255611875452571, 0.723388848660391, 0.724112961622013, 0.7255611875452571, 0.7255611875452571, 0.7262853005068791, 0.7255611875452571, 0.723388848660391, 0.721940622737147, 0.722664735698769, 0.723388848660391, 0.719768283852281, 0.720492396813903, 0.719044170890659, 0.7168718320057929, 0.719044170890659, 0.719044170890659, 0.722664735698769, 0.724112961622013, 0.723388848660391, 0.718320057929037, 0.721216509775525]

```
best K is 3
Custom Decision Tree - Train Accuracy: 0.97, Test Accuracy: 0.91
Custom Decision Tree - Train Time: 8.06 seconds, Test Time: 0.02 seconds

Custom KNN - Train Accuracy: 0.90, Test Accuracy: 0.78
Custom KNN - Train Time: 0.00 seconds, Test Time: 4.43 seconds

PS C:\Users\ASUS\Desktop>
```

which gives us the following table

	Train Accuracy	Test Accuracy
Custom Decision Tree	0.97	0.91
Custom KNN	0.90	0.78