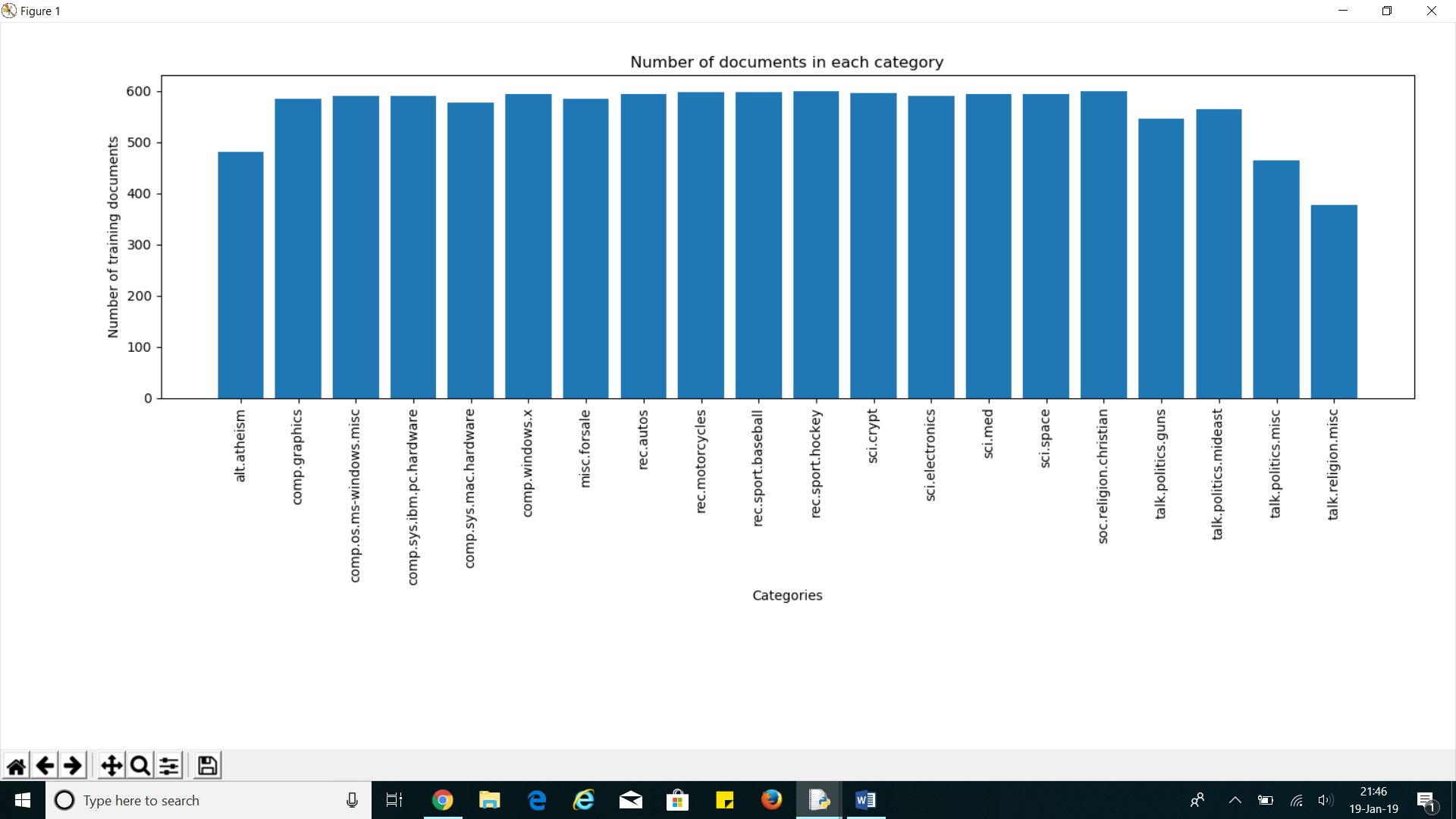
**Kushagra Rastogi**

**304640248**

**ECE 219**

**Project 1: Classification Analysis on Textual Data**

**QUESTION 1:**



It can be seen that the number of documents, for the most part, are roughly evenly distributed.

**QUESTION 2:**

Shape of TF-IDF train matrix = (4732, 17671)

Shape of TF-IDF test matrix = (3150, 17671)

**QUESTION 3:**

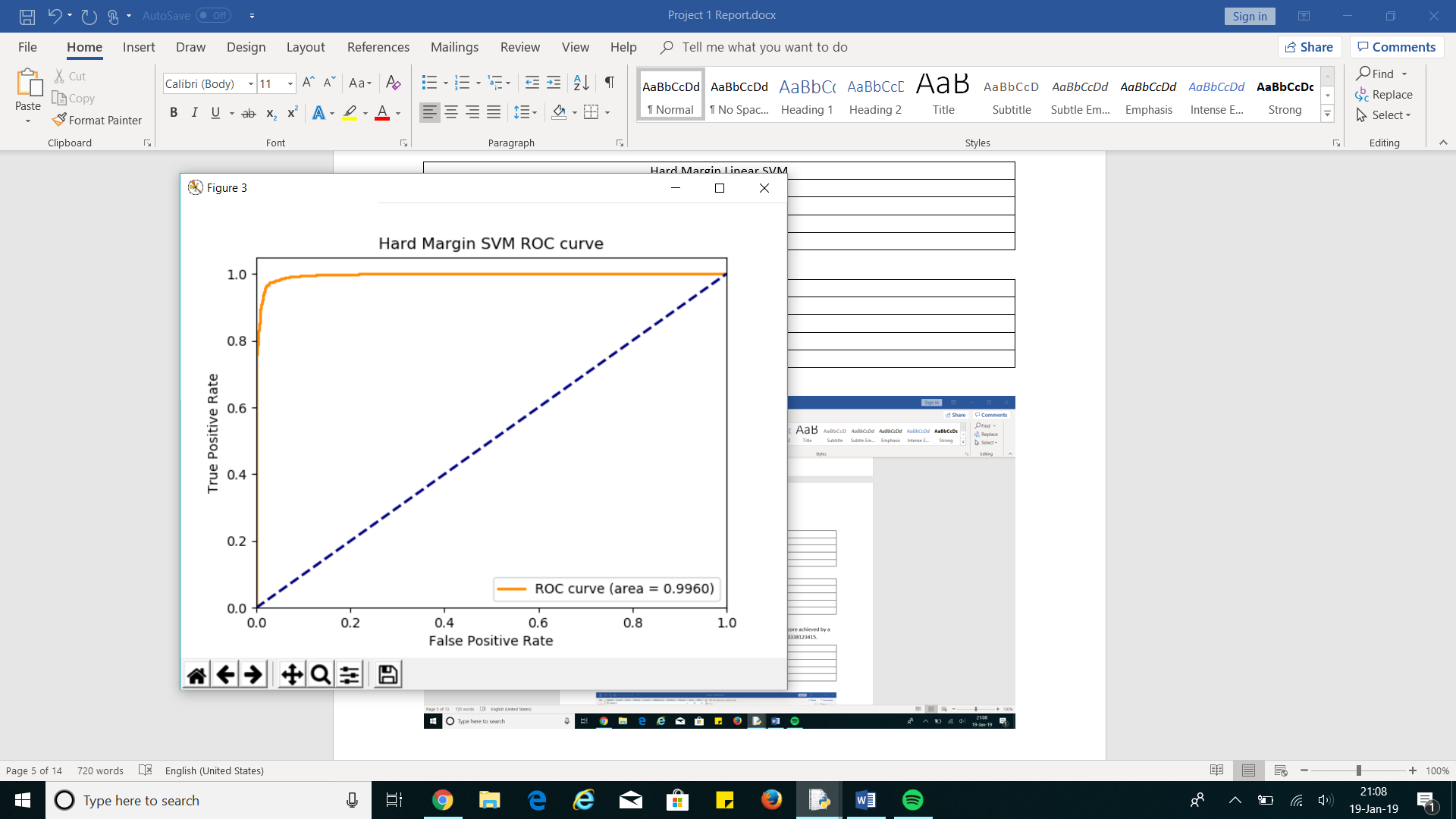
Error for LSI = 4116.55861302211

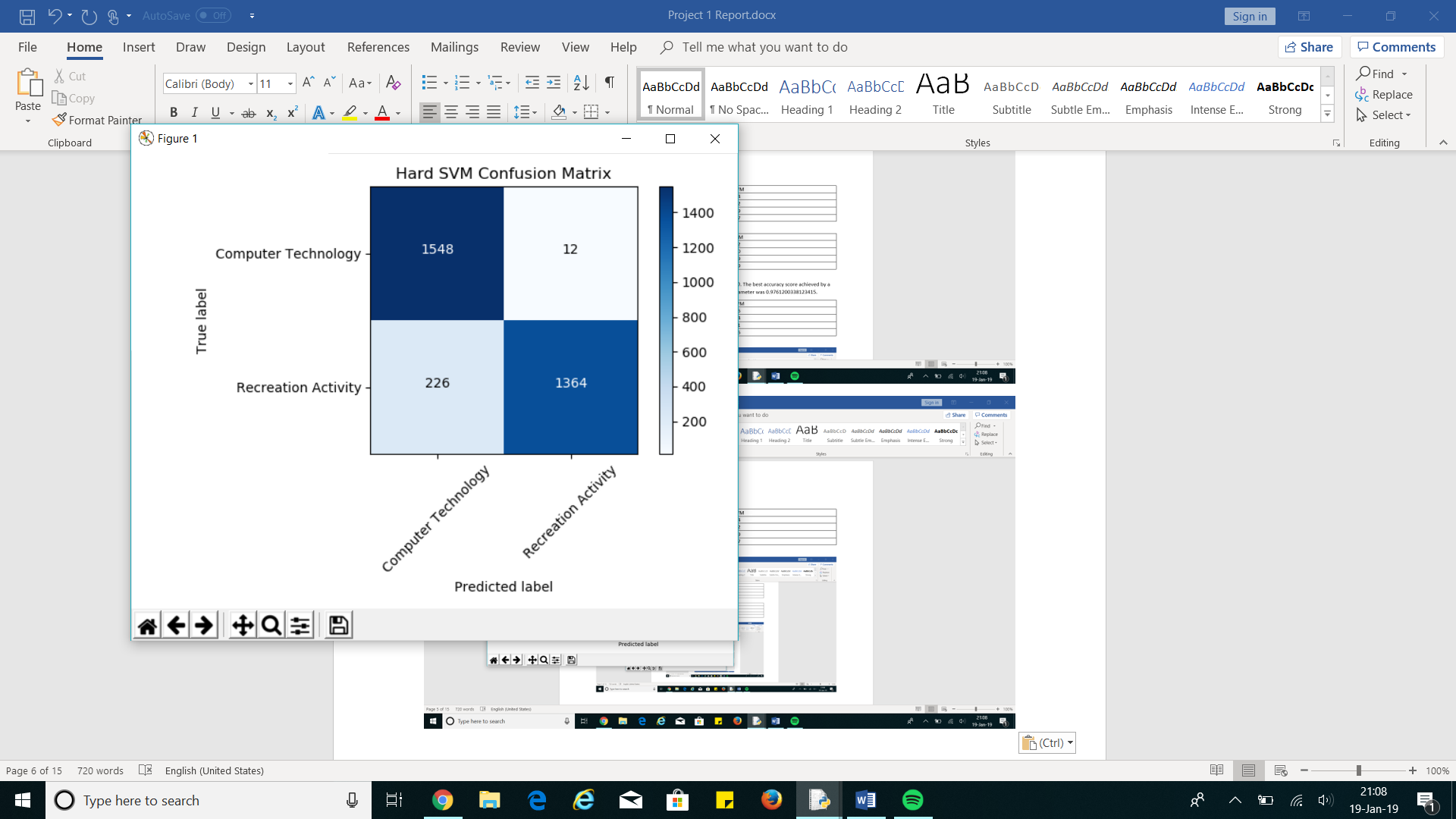
Error for NMF = 4154.1833496899435

Thus, the error for NMF is larger. This is because the NMF optimization objectives have more constraints than LSI. The added constraints are that and . This reduces the degrees of freedom and makes the search space more restrictive. On the other hand, LSI has a much larger search space to explore. This means that it has more freedom to find the best vectors that minimize the error and the best vectors can include negative elements. This is not possible in NMF.

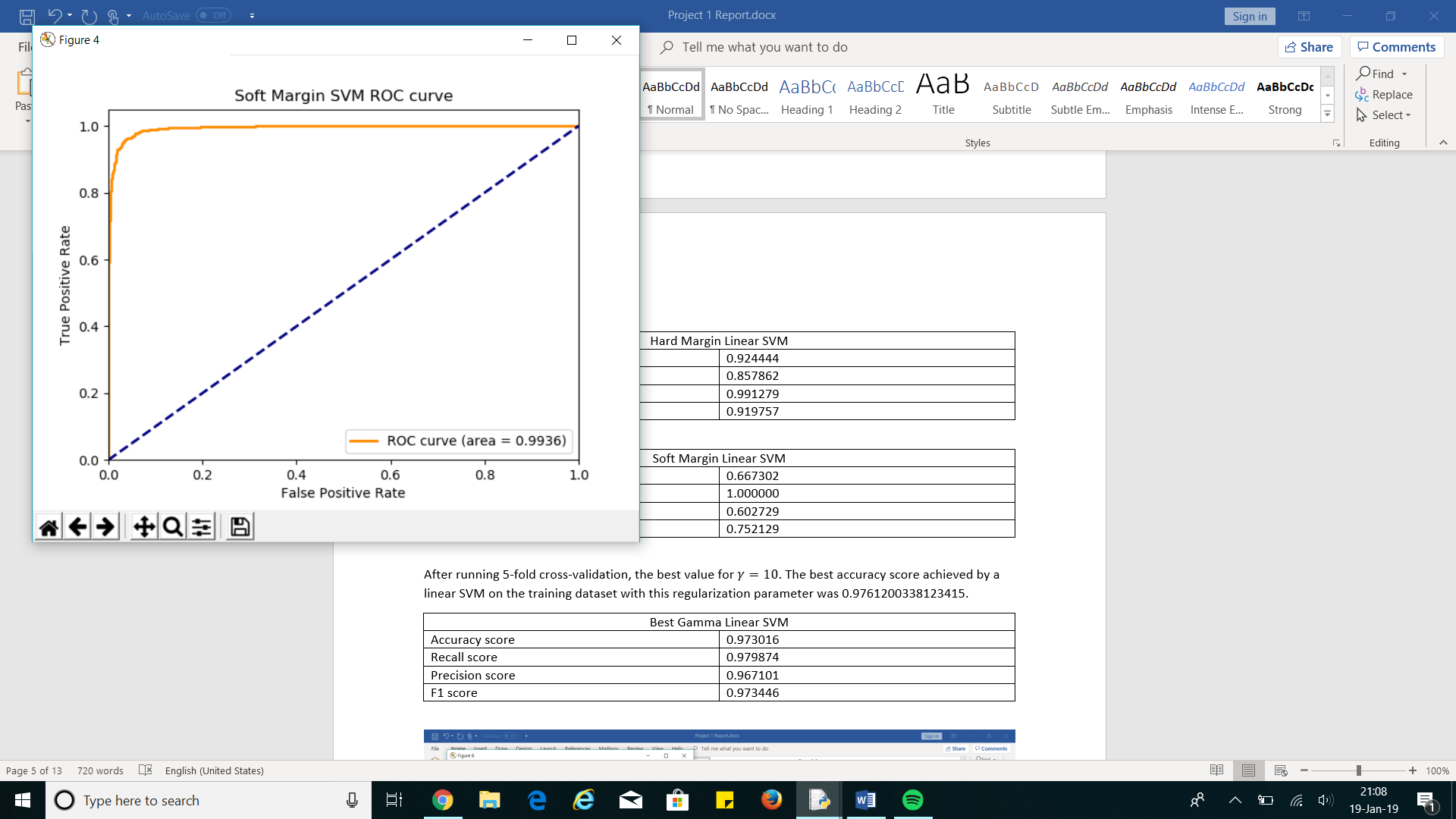
**QUESTION 4:**

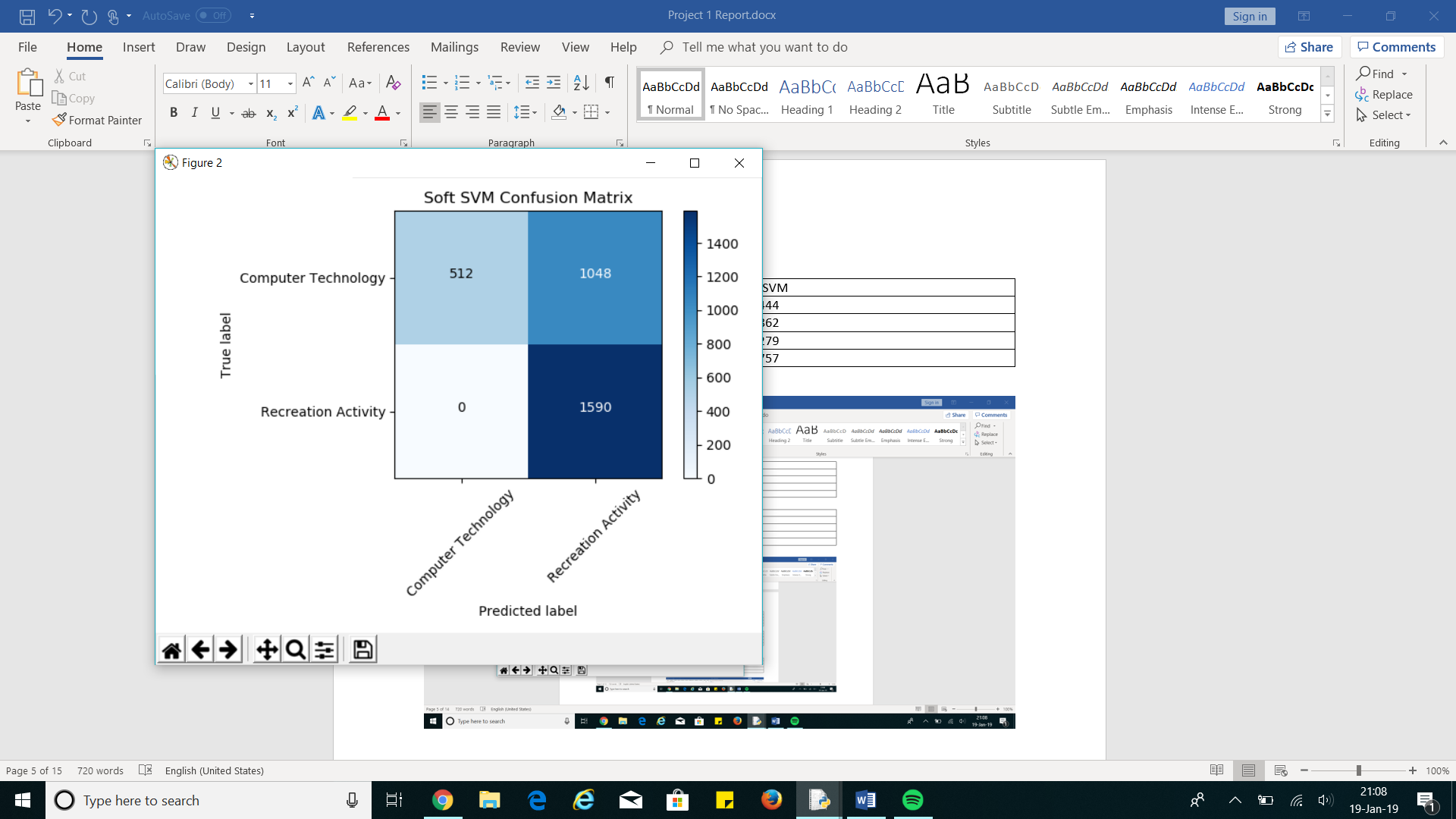
|  |  |
| --- | --- |
| Hard Margin Linear SVM | |
| Accuracy score | 0.924444 |
| Recall score | 0.857862 |
| Precision score | 0.991279 |
| F1 score | 0.919757 |





|  |  |
| --- | --- |
| Soft Margin Linear SVM | |
| Accuracy score | 0.667302 |
| Recall score | 1.000000 |
| Precision score | 0.602729 |
| F1 score | 0.752129 |

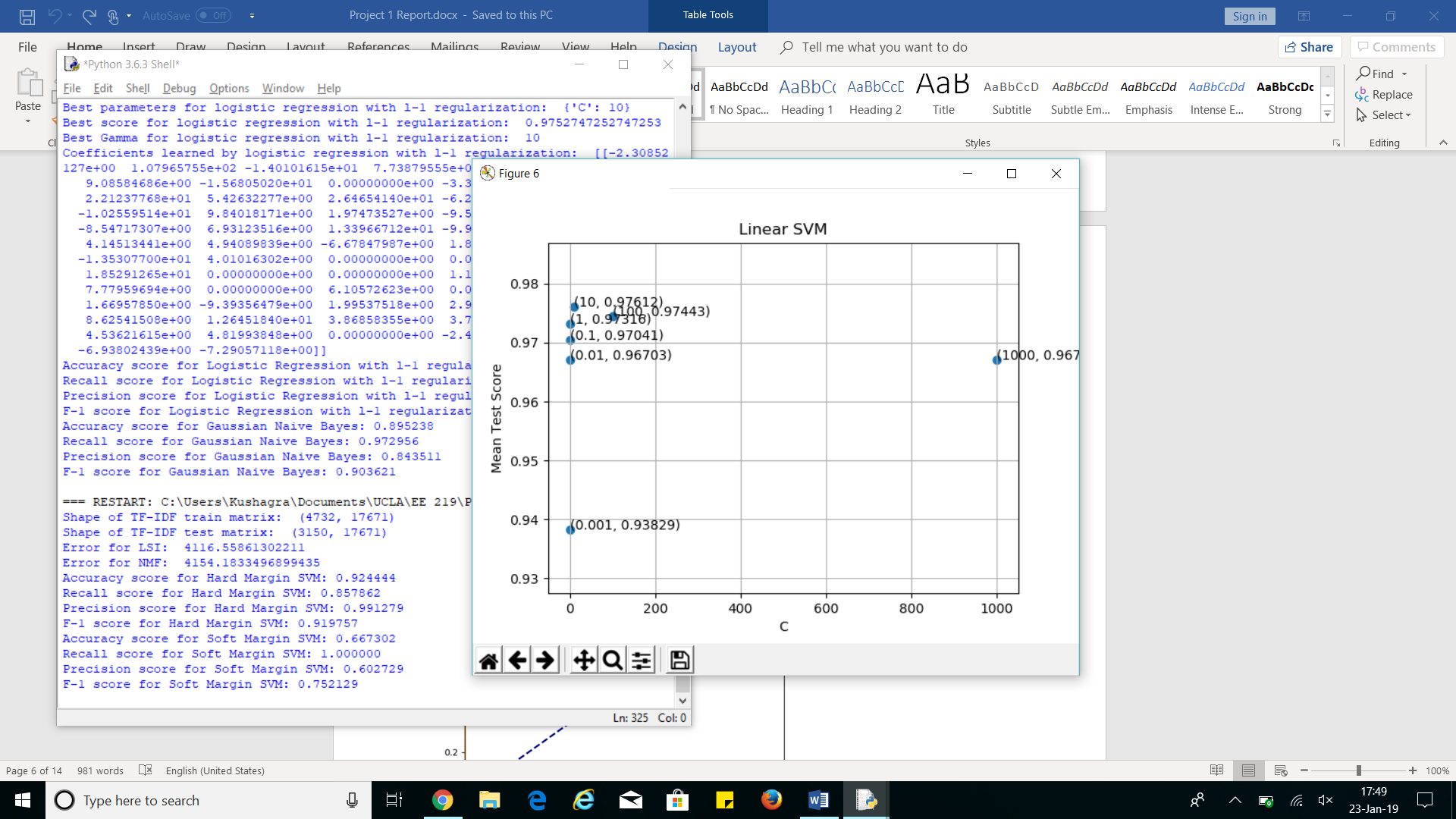




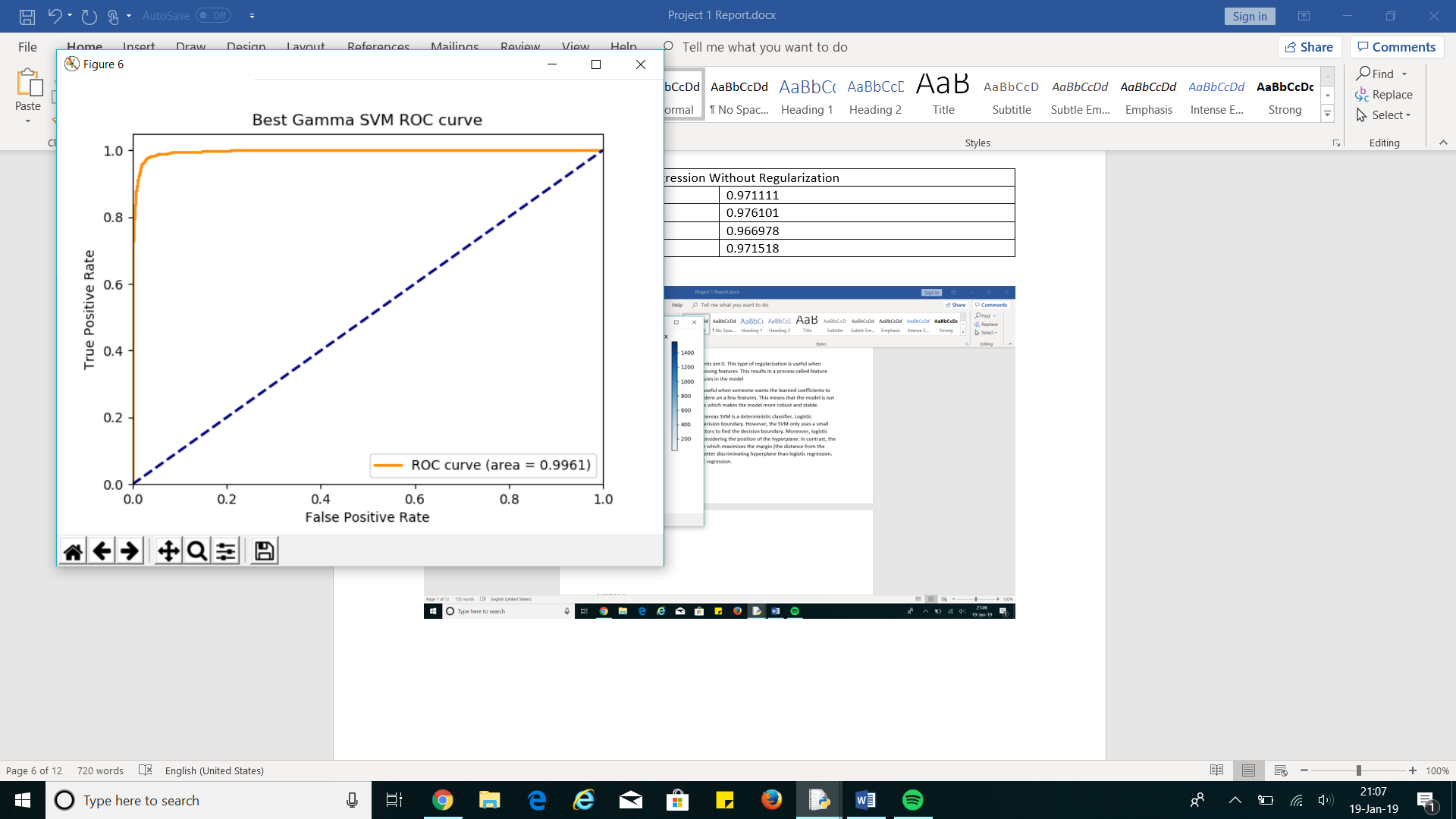
The Hard Margin SVM performs better than Soft Margin SVM.

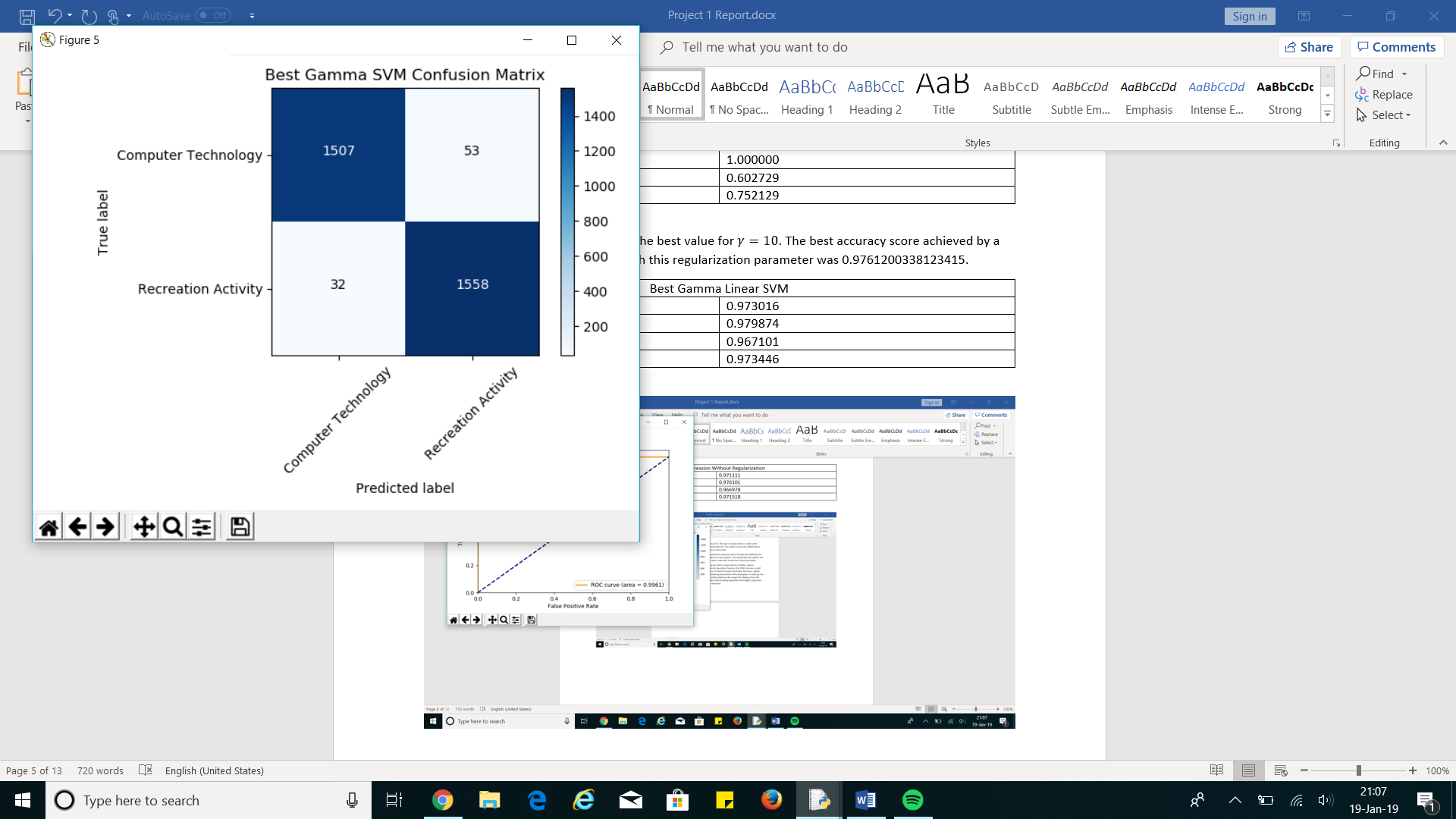
The ROC curve for the Soft Margin SVM and recall score look good. This conflicts with the other metrics which are in the range of 0.6 – 0.75. This might be because the Soft Margin SVM has found a separating hyperplane whose intercept term is not adjusted properly. Thus, the SVM may have found the optimal vector but not the term. This means that the margin may not be fully maximized.

After running 5-fold cross-validation, the best value for . The best accuracy score achieved by a linear SVM on the training dataset with this regularization parameter was 0.9761200338123415.



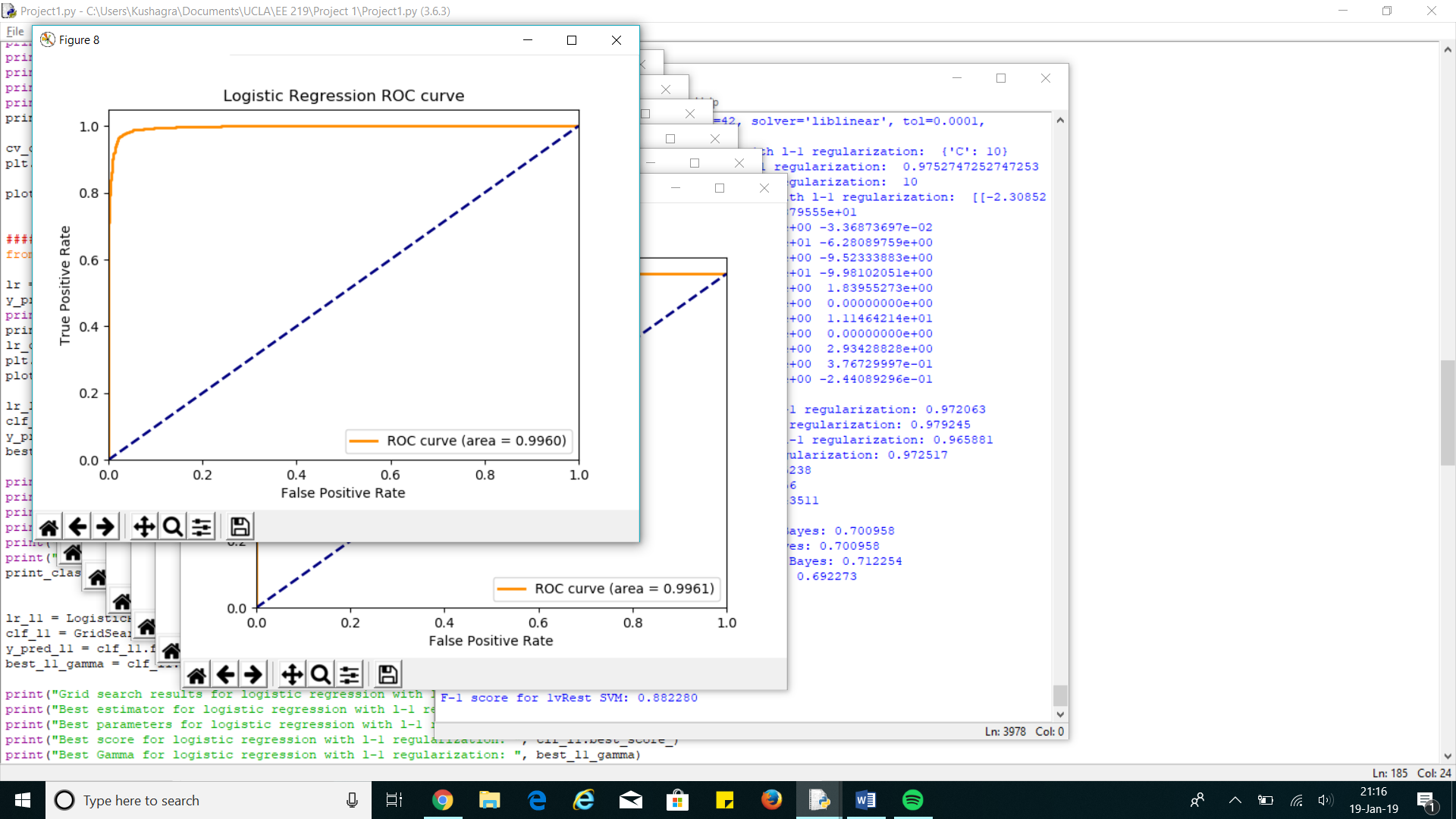
|  |  |
| --- | --- |
| Best Gamma Linear SVM () | |
| Accuracy score | 0.973016 |
| Recall score | 0.979874 |
| Precision score | 0.967101 |
| F1 score | 0.973446 |

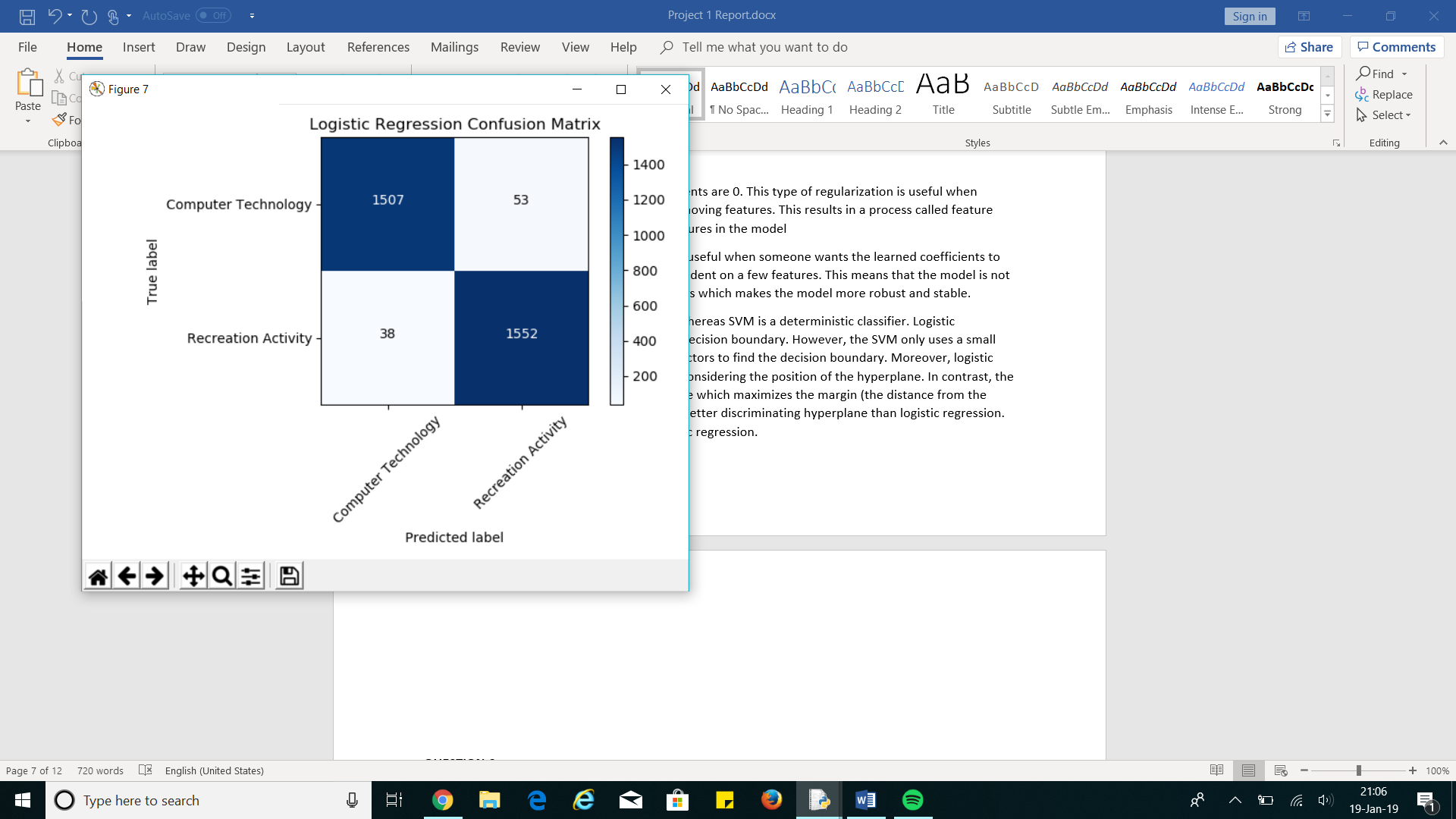




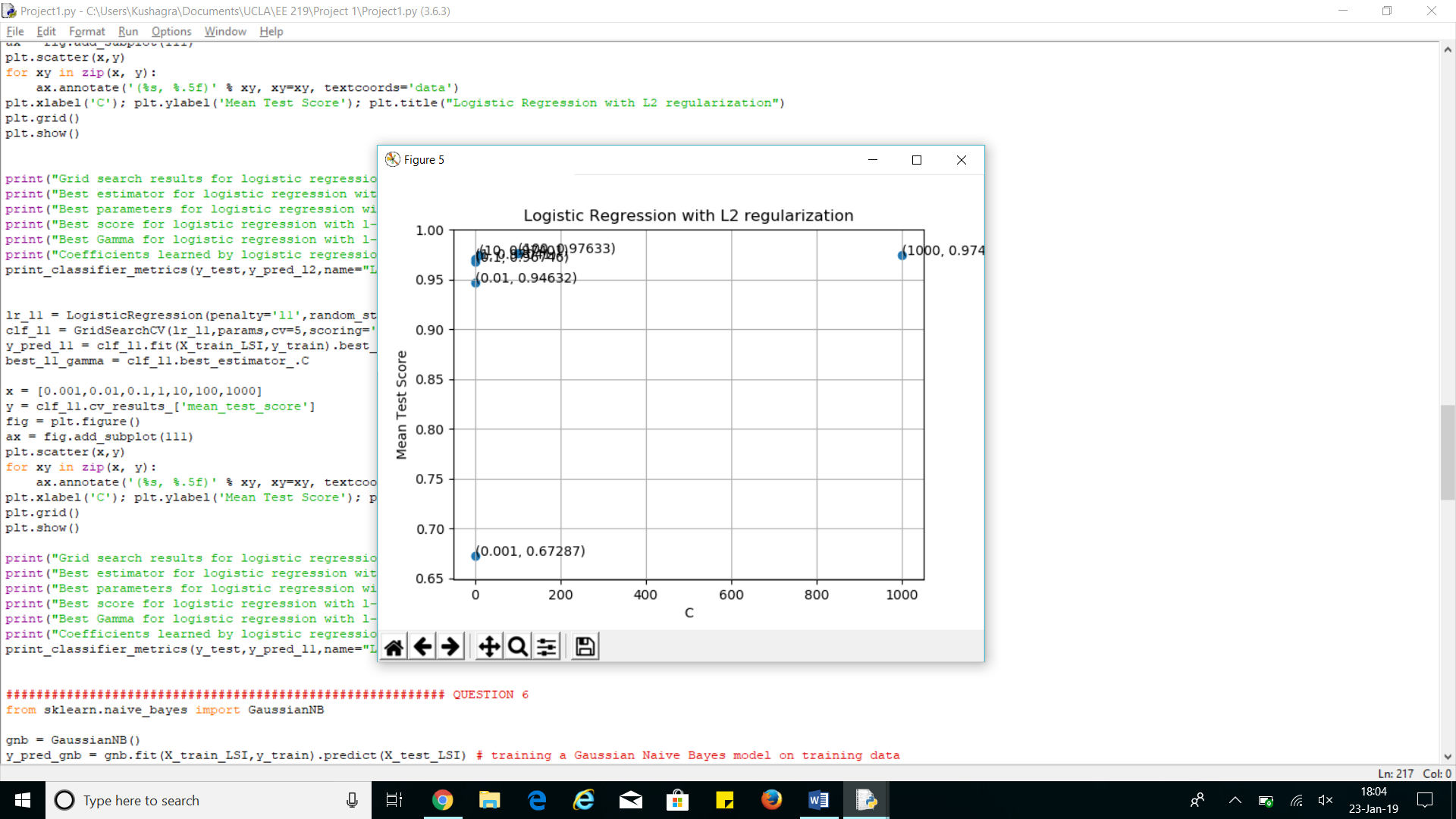
**QUESTION 5:**

|  |  |
| --- | --- |
| Logistic Regression Without Regularization | |
| Accuracy score | 0.971111 |
| Recall score | 0.976101 |
| Precision score | 0.966978 |
| F1 score | 0.971518 |



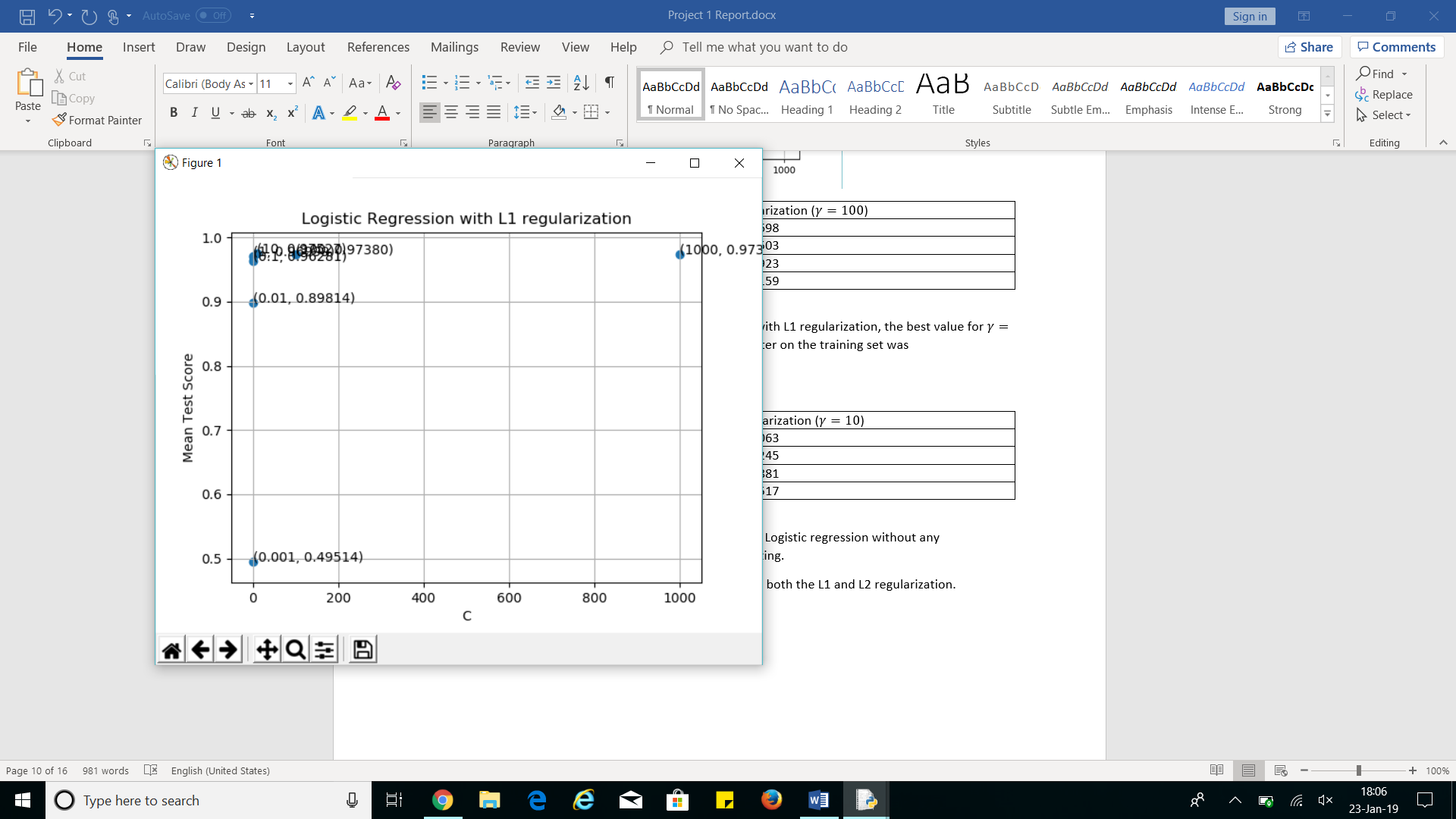


After running 5-fold cross-validation for logistic regression with L2 regularization, the best value of . The best accuracy score with this regularization parameter on the training set was 0.9763313609467456.



|  |  |
| --- | --- |
| Logistic Regression With L2 Regularization () | |
| Accuracy score | 0.972698 |
| Recall score | 0.980503 |
| Precision score | 0.965923 |
| F1 score | 0.973159 |

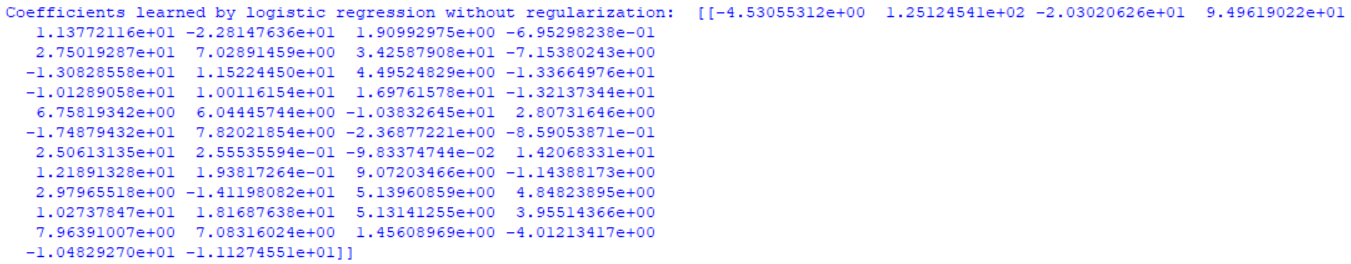
After running 5-fold cross-validation for logistic regression with L1 regularization, the best value for . The best accuracy score with this regularization parameter on the training set was 0.9752747252747253.

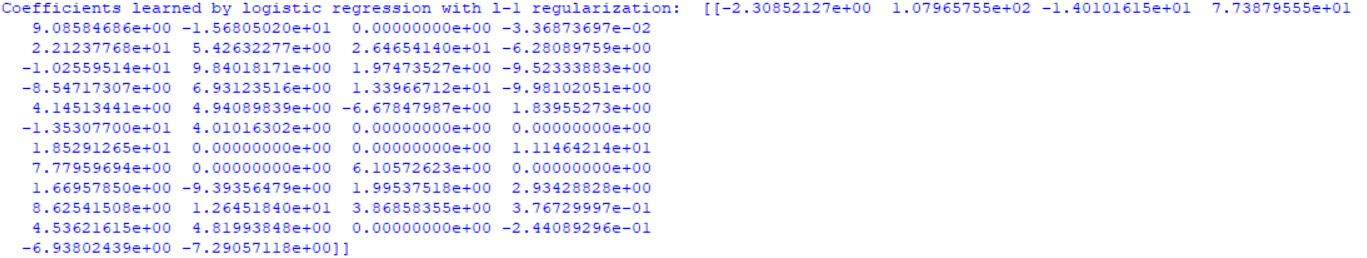


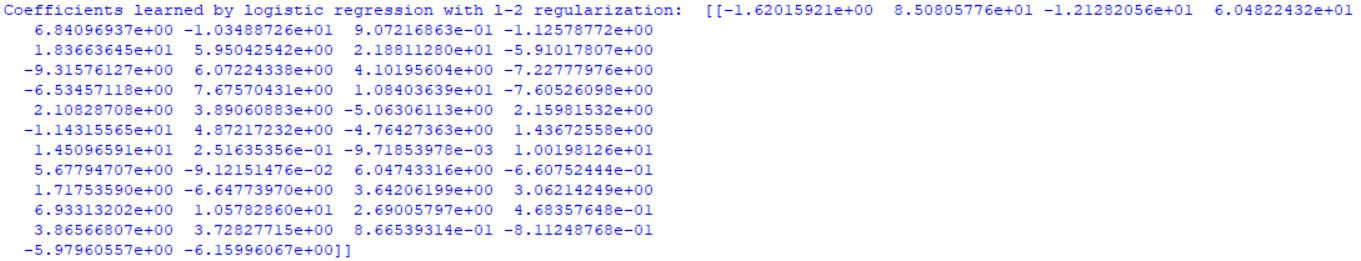
|  |  |
| --- | --- |
| Logistic Regression With L1 Regularization () | |
| Accuracy score | 0.972063 |
| Recall score | 0.979245 |
| Precision score | 0.965881 |
| F1 score | 0.972517 |

Logistic regression with L2 regularization performs the best because it has the highest accuracy and F1 score. Logistic regression without any regularization performs the worst because it scores the lowest scores in all categories. Logistic regression with L1 regularization performs satisfactorily. It is better than logistic regression without regularization but worse than logistic regression with L2 regularization.

The regularization parameter decreases the test error in the both the L1 and L2 regularization.

No regularization: The learned coefficients tend to be quite large, usually on the order of magnitude of . This makes the model volatile which is a sign of over-fitting. This type of regularization is useful when someone wants to build a complex model to deal with complex data and there are no signs of overfitting.

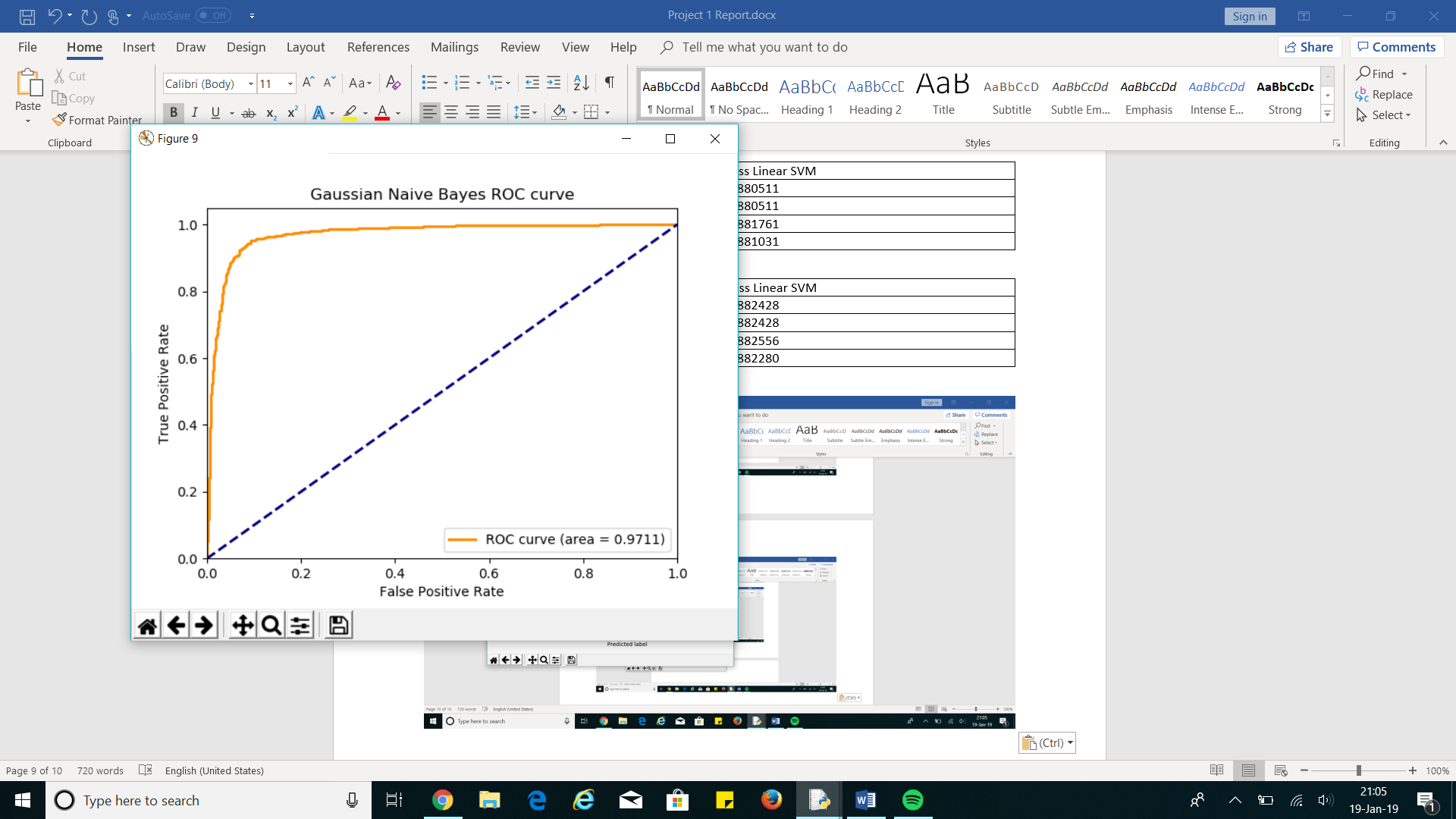
L1 regularization: Some of the learned coefficients are 0 and most of the learned coefficients are on the order of magnitude of which is smaller than the learned coefficients of logistic regression without any regularization. This type of regularization is useful when someone wants to build a sparse model by removing features. This results in a process called feature selection which keeps the most significant features in the model

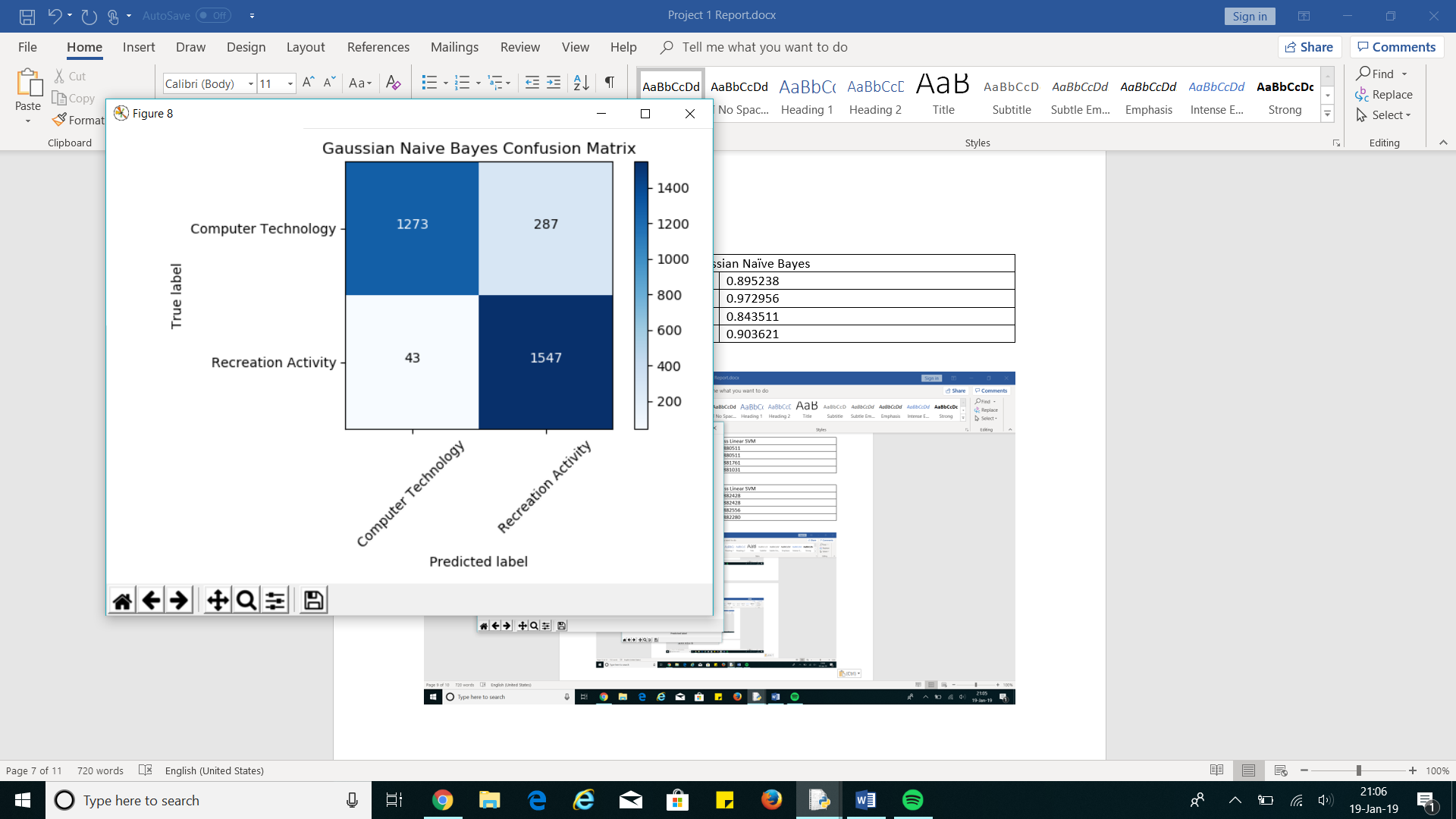
L2 regularization: Most of the learned coefficients are on the order of magnitude of which is smaller than the learned coefficients of logistic regression without any regularization. This type of regularization is useful when someone wants the learned coefficients to be small so that the model is not heavily dependent on a few features. This means that the model is not sensitive to small changes in the feature vectors which makes the model more robust and stable.

Logistic regression is a probabilistic classifier whereas SVM is a deterministic classifier. Logistic regression uses all the data points to find the decision boundary. However, the SVM only uses a small subset of the data points called the support vectors to find the decision boundary. Moreover, logistic regression finds a decision boundary without considering the position of the hyperplane. In contrast, the SVM tries to find the position of the hyperplane which maximizes the margin (the distance from the support vectors). In this way, the SVM finds a better discriminating hyperplane than logistic regression. Thus, SVM has better performance than logistic regression.

**QUESTION 6:**

|  |  |
| --- | --- |
| Multiclass Gaussian Naïve Bayes | |
| Accuracy score | 0.895238 |
| Recall score | 0.972956 |
| Precision score | 0.843511 |
| F1 score | 0.903621 |





**QUESTION 7:**

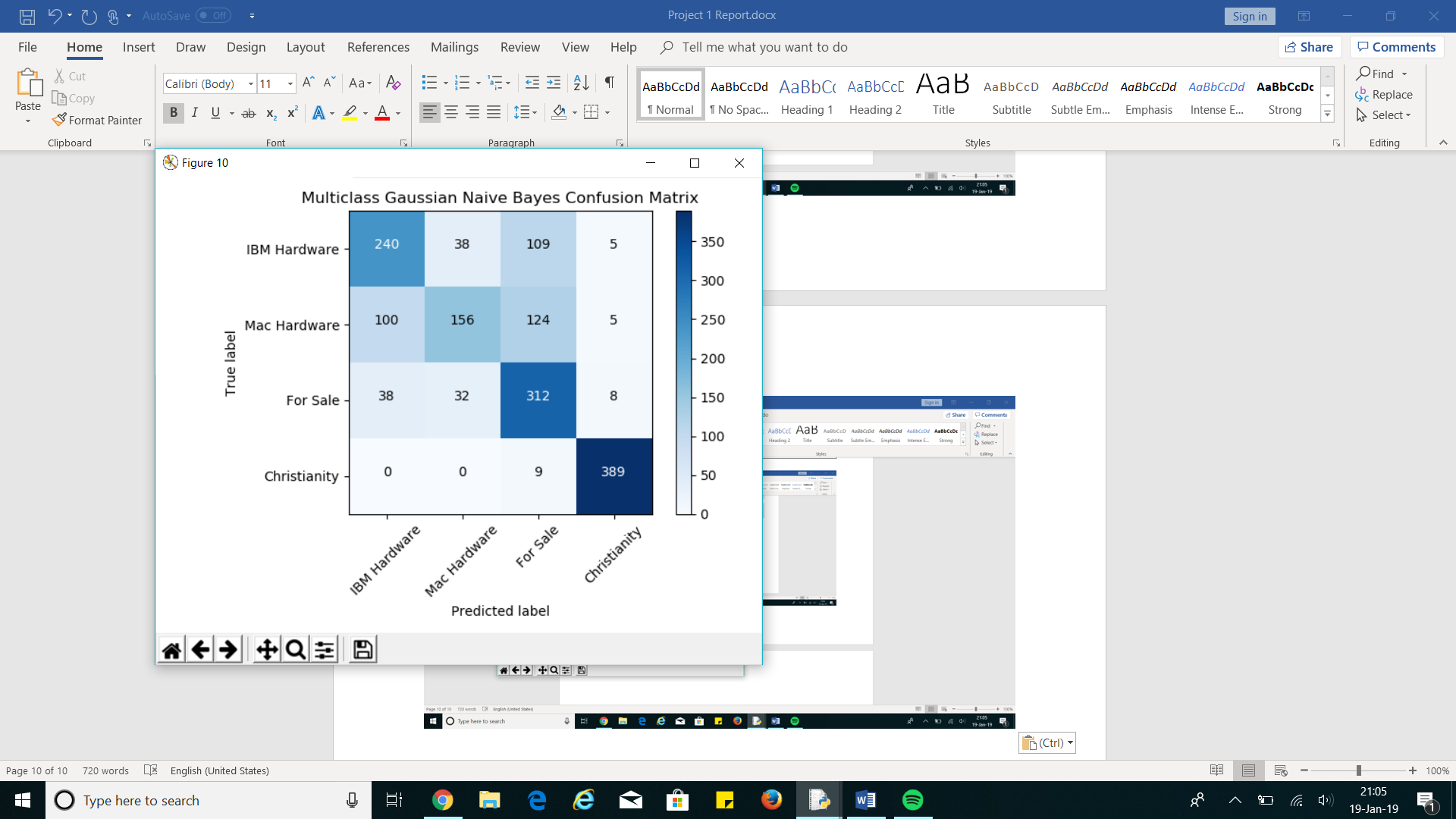
|  |  |  |
| --- | --- | --- |
| Pipeline | Best Combination after running GridSearchCV | Test Accuracy Score |
| 1 | Remove headers and footers  +  Lemmatization  +  min\_df = 3  +  LSI  +  Logistic Regression () | 0.965397 |
| 2 | Keep headers and footers  +  Lemmatization  +  min\_df = 3  +  LSI  +  Linear SVM () | 0.973016 |

Thus, the best combination is Pipeline 2.

**QUESTION 8:**

Since LSI provides a better low-dimensional representation than NMF, I decided to stick with Gaussian Naïve Bayes instead of using Multinomial Naïve Bayes. After experimenting with several metric settings, I found out that average= ‘weighted’ for precision, recall and F1 score gives the best results. Also, the ‘weighted’ setting takes label imbalance into account which provides a more informed score. This is not a problem in this dataset since the labels are roughly evenly distributed.

|  |  |
| --- | --- |
| Multiclass Gaussian Naïve Bayes | |
| Accuracy score | 0.700958 |
| Recall score | 0.700958 |
| Precision score | 0.712254 |
| F1 score | 0.692273 |



After running 5-fold grid search cross-validation, the best regularization parameter was .

|  |  |
| --- | --- |
| One VS One Multiclass Linear SVM with | |
| Accuracy score | 0.880511 |
| Recall score | 0.880511 |
| Precision score | 0.881761 |
| F1 score | 0.881031 |

After running 5-fold grid search cross-validation, the best regularization parameter was .

|  |  |
| --- | --- |
| One VS Rest Multiclass Linear SVM with | |
| Accuracy score | 0.889457 |
| Recall score | 0.889457 |
| Precision score | 0.889584 |
| F1 score | 0.889282 |

