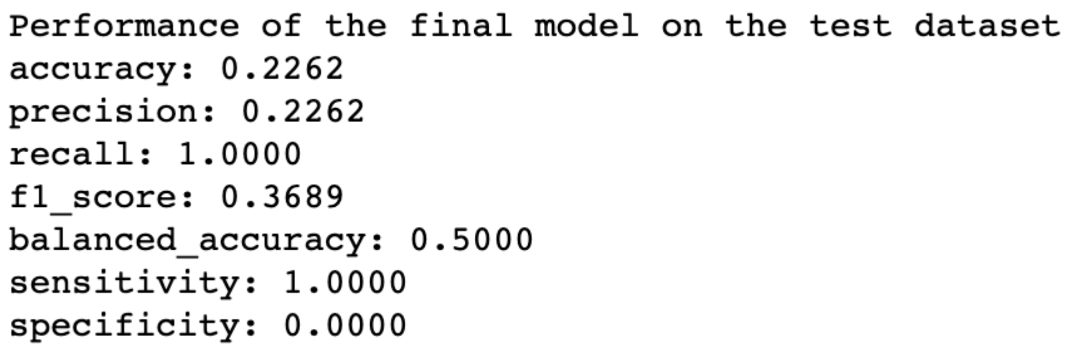
ASSIGNMENT 3 REPORT

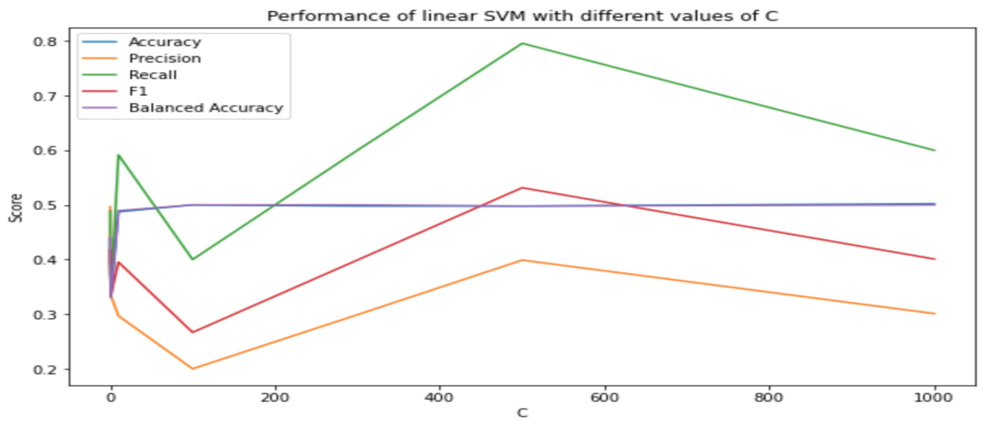
QUESTION 1:

The performance of the final model on the test dataset is not very good, with an ***accuracy score of 0.7738***. However, ***the precision, recall, and F1 score are all 0***, which suggests that the model did not correctly identify any positive instances. The ***balanced accuracy score is 0.5***, which suggests that the model's performance is poor. A ***sensitivity of 1.0000*** meaning that the model correctly predicted all the positive cases in the test set and ***specificity of 0.0000*** meaning that the model did not correctly predict any of the negative cases in the test set. This suggests that the model may be overfitting to the positive cases in the training set, and is not generalizing well.

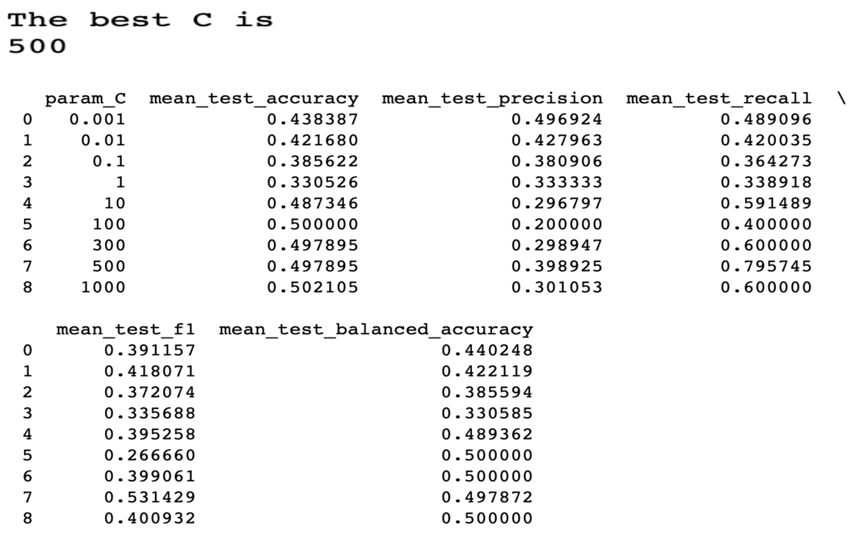


```
Performance of the final model on the test dataset
accuracy: 0.2262
precision: 0.2262
recall: 1.0000
f1_score: 0.3689
balanced_accuracy: 0.5000
sensitivity: 1.0000
specificity: 0.0000
```

The table and the figure below shows ***the performance metrics for different values of the hyperparameter C during hyperparameter tuning***. As the value of C increases, the model tends to overfit the data, which we can find from the decreasing value in accuracy, precision, recall, f1_score, and balanced accuracy after a certain value of C.



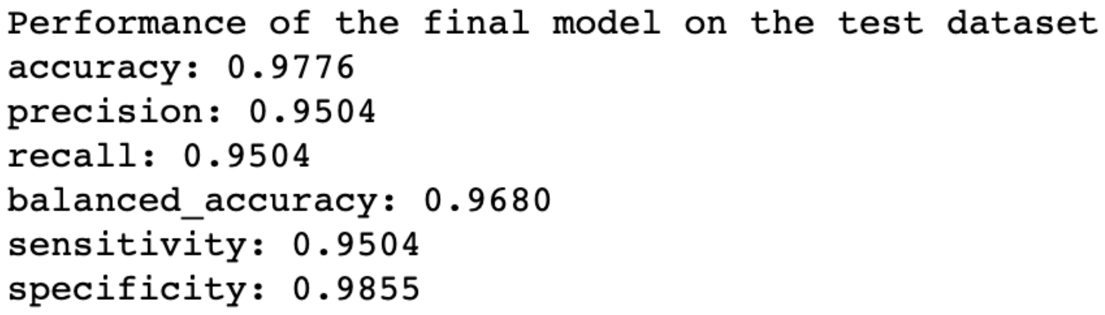
For small values of C (0.001 and 0.01), the model performs relatively better. As the value of C increases to 0.1, the model's performance starts decreasing, and it becomes worse as C increases further. However, the model's performance suddenly improves when C is set to 10 and continues and reaches a good place when C is around 500 and then drops in performance which could be due to overfitting.



```
The best C is
500

  param_C  mean_test_accuracy  mean_test_precision  mean_test_recall  \
0   0.001            0.438387             0.496924          0.489096
1    0.01            0.421680             0.427963          0.420035
2     0.1            0.385622             0.380906          0.364273
3       1            0.330526             0.333333          0.338918
4      10            0.487346             0.296797          0.591489
5     100            0.500000             0.200000          0.400000
6     300            0.497895             0.298947          0.600000
7     500            0.497895             0.398925          0.795745
8    1000            0.502105             0.301053          0.600000

   mean_test_f1  mean_test_balanced_accuracy
0      0.391157                     0.440248
1      0.418071                     0.422119
2      0.372074                     0.385594
3      0.335688                     0.330585
4      0.395258                     0.489362
5      0.266660                     0.500000
6      0.399061                     0.500000
7      0.531429                     0.497872
8      0.400932                     0.500000
```

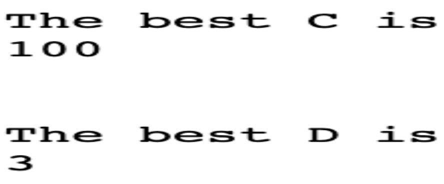
QUESTION 2:

The final model achieved a high level of performance on the test dataset with an ***accuracy of 0.9776***, indicating that 97.76% of the test data was correctly classified. The ***precision and recall were both 0.9504***, which means that 95.04% of the predicted positive cases were true positives and 95.04% of the actual positive cases were correctly identified by the model, respectively. The ***f1_score was also 0.9504,*** indicating a good balance. The model's ***balanced accuracy was 0.9680***, which suggests that it performed well in classifying both the positive and negative cases, without being biased towards one class. Also the sensitivity and specificity values of 0.9504 and 0.9855, respectively, suggest that the model has good performance in correctly identifying positive cases and negative cases, respectively.



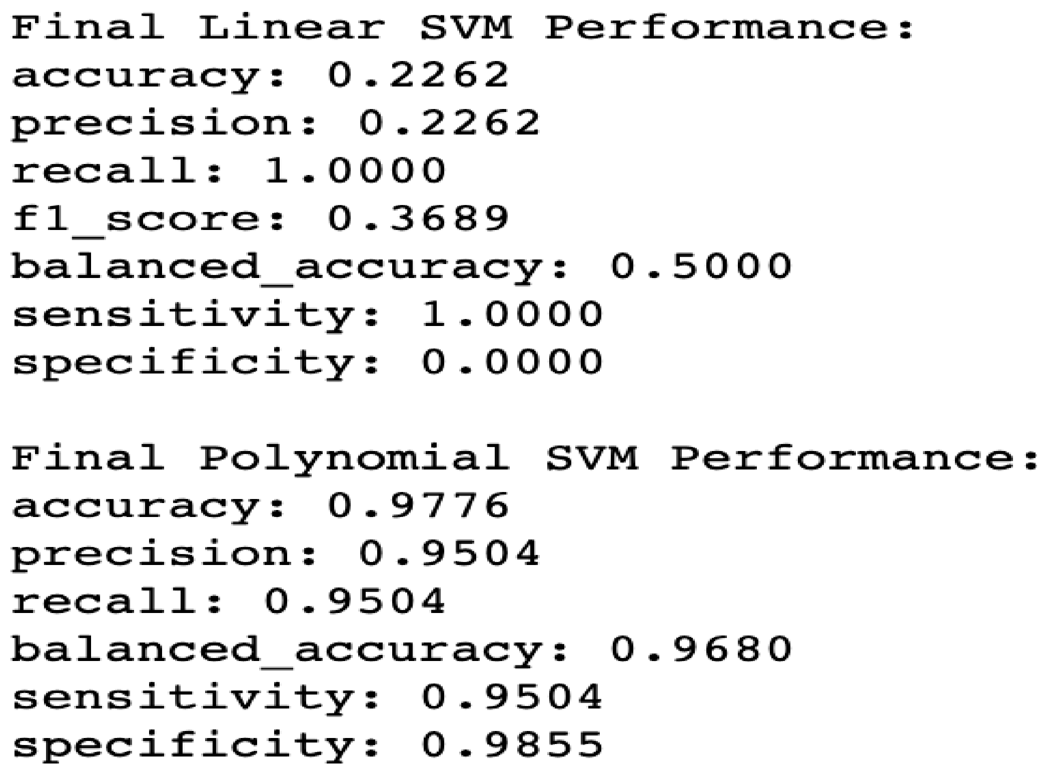
```
Performance of the final model on the test dataset
accuracy: 0.9776
precision: 0.9504
recall: 0.9504
balanced_accuracy: 0.9680
sensitivity: 0.9504
specificity: 0.9855
```

Similarly, the best C and the d achieved during the tuning is as follows.



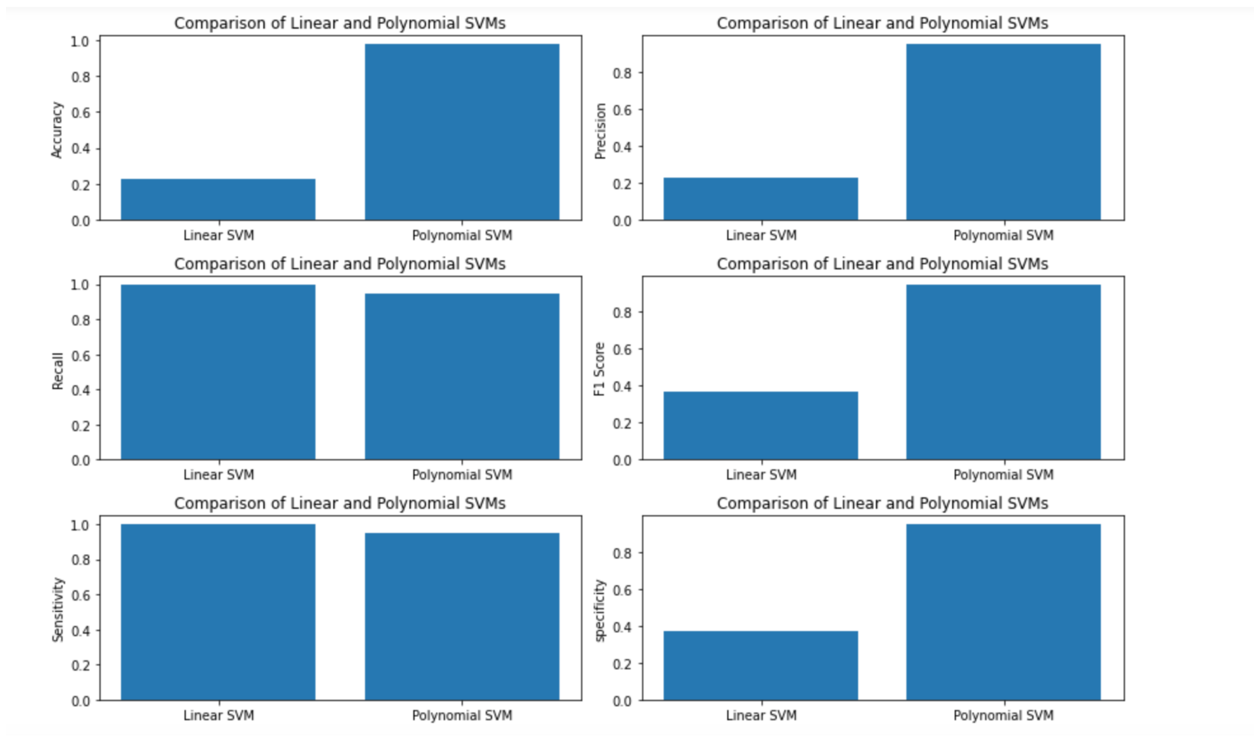
```
The best C is
100


The best D is
3
```

Comparing the performance of the final polynomial kernel SVM model and the final linear SVM model, it is evident that the former outperforms the latter. The ***accuracy*** of the ***final polynomial SVM model is 0.9776***, which is ***significantly higher*** than the accuracy of the final linear SVM model, which is only ***0.7738***.



```
Final Linear SVM Performance:
accuracy: 0.2262
precision: 0.2262
recall: 1.0000
f1_score: 0.3689
balanced_accuracy: 0.5000
sensitivity: 1.0000
specificity: 0.0000

Final Polynomial SVM Performance:
accuracy: 0.9776
precision: 0.9504
recall: 0.9504
balanced_accuracy: 0.9680
sensitivity: 0.9504
specificity: 0.9855
```

Additionally, ***the precision and recall of the final polynomial SVM model are all higher*** than those of the final linear SVM model. One notable difference between the two models is the balanced accuracy, which is a metric that takes into account the imbalance in the target classes. The balanced accuracy of the final polynomial SVM model is 0.9680, which is also higher than that of the final linear SVM model, which is only 0.5000. This indicates that the final polynomial SVM model is better able to handle the class imbalance in the dataset. Similarly the sensitivity and specificity of the final polynomial is much better than the linear SVM.



Overall, these results demonstrate that the polynomial kernel SVM model is a more effective approach for this particular classification problem than the linear SVM model.

QUESTION 3:

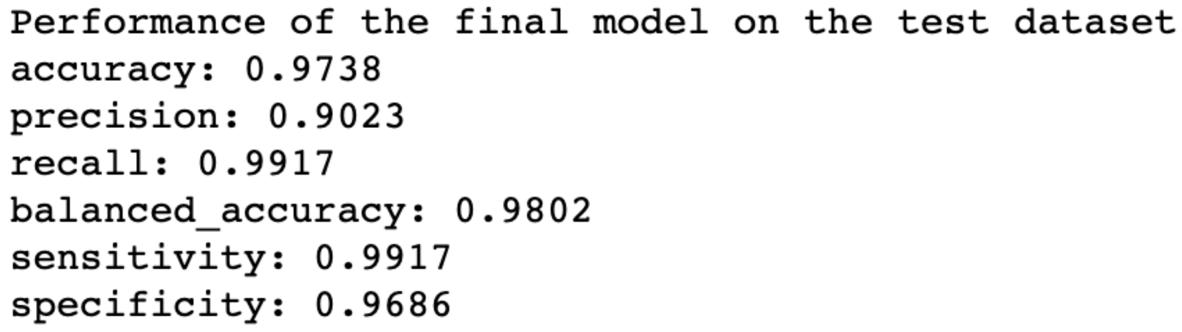
The best C and the gamma that we found are as follows

```
The best C is
10


The best gamma is
10
```

Analyzing the performance metrics (shown below) suggest that the final model has performed well on the test dataset.

- ***Accuracy:*** 0.9738 indicates that 97.38% of the predictions made by the model were correct.
- ***Precision:*** 0.9023 indicates that 90.23% of the positive predictions made by the model were correct.
- ***Recall:*** 0.9917 indicates that 99.17% of the actual positives in the test dataset were correctly predicted by the model.

- ***Balanced accuracy:*** 0.9802 indicates the average of the recall obtained on each class.
- ***Sensitivity:*** A sensitivity value of 0.9917 means that the model correctly identified 99.17% of the positive cases.
- ***Specificity:*** A Specificity value of 0.9686 means that the model correctly identified 96.86% of the negative cases.



```
Performance of the final model on the test dataset
accuracy: 0.9738
precision: 0.9023
recall: 0.9917
balanced_accuracy: 0.9802
sensitivity: 0.9917
specificity: 0.9686
```

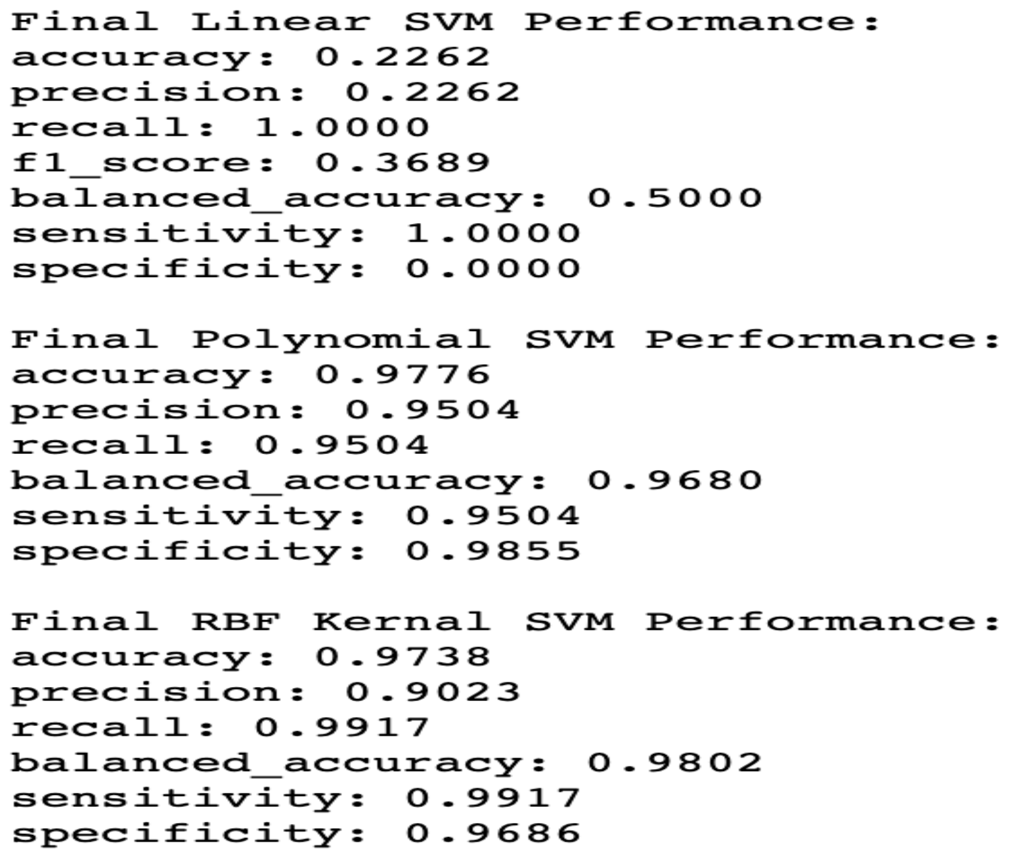
Overall, the model seems to have performed well, achieving high scores for both precision and recall, indicating that the model is both accurate and effective at identifying the positive class. Additionally, the balanced accuracy score suggests that the model is capable of generalizing well to new data, even when the classes are imbalanced.

Comparing the performance of the three models we created so far

The linear SVM model has a low accuracy of 0.7738, and the precision, recall, and F1 score are all 0, indicating that the model is not making any positive predictions. This may be due to the linear kernel being too simple for this dataset, or the model may not have been properly trained.

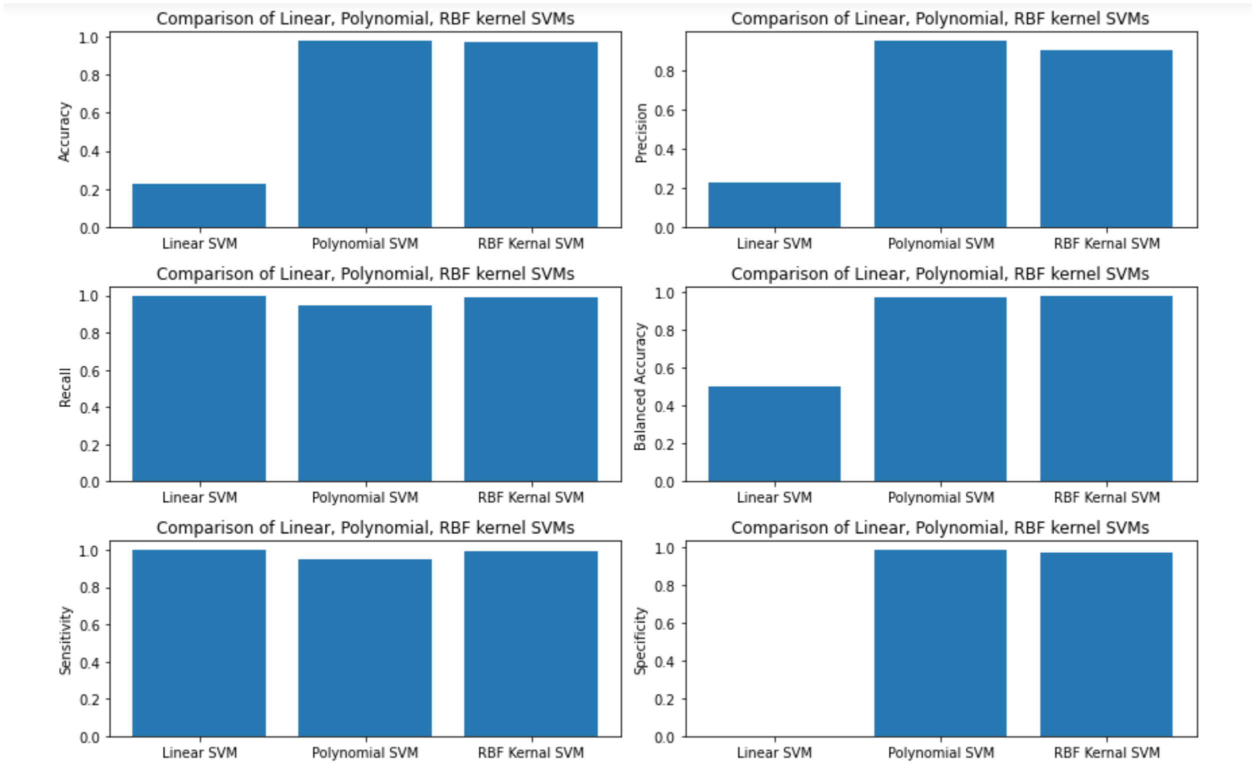
The polynomial SVM model has a higher accuracy of 0.9776, with a precision, recall, and F1 score of 0.9504. This indicates that the model is performing well and making accurate predictions. The balanced accuracy is also high at 0.9680, indicating that the model is not biased towards one class over the other.

The RBF kernel SVM model has similar performance to the polynomial SVM model, with an accuracy of 0.9738, precision of 0.9023, recall of 0.9917, F1 score of 0.9504, and a high balanced accuracy of 0.9802. The precision is not as high as the polynomial SVM model, which suggests that the RBF kernel may be causing the model to make more false positives.



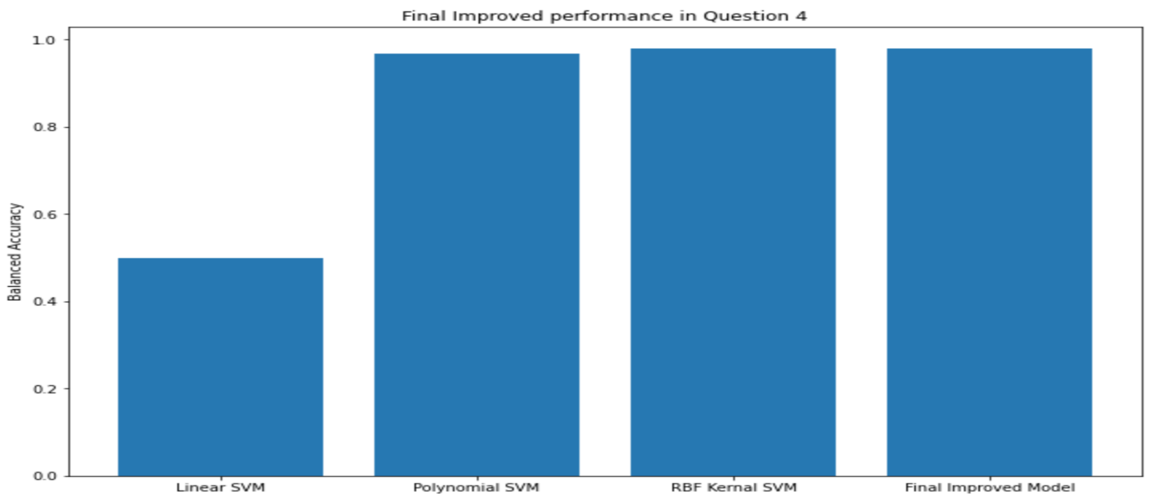
```
Final Linear SVM Performance:
accuracy: 0.2262
precision: 0.2262
recall: 1.0000
f1_score: 0.3689
balanced_accuracy: 0.5000
sensitivity: 1.0000
specificity: 0.0000

Final Polynomial SVM Performance:
accuracy: 0.9776
precision: 0.9504
recall: 0.9504
balanced_accuracy: 0.9680
sensitivity: 0.9504
specificity: 0.9855

Final RBF Kernal SVM Performance:
accuracy: 0.9738
precision: 0.9023
recall: 0.9917
balanced_accuracy: 0.9802
sensitivity: 0.9917
specificity: 0.9686
```

Overall, it seems that ***the polynomial SVM model performed the best out of the three models***.



QUESTION 4:

Since polynomial kernel SVM performed best, we have used the same with the best C value and degree. The null values in the dataframe are handled by filling them with the mean value of the respective column and outliers are removed using Z-score. We have scaled the feature using standard scalar. Then we have trained and tested the data which ***improved the balanced accuracy by 0.002***.


Final Improved performance in Question 4
Balanced Accuracy
Linear SVM
Polynomial SVM
RBF Kernal SVM
Final Improved Model