```
1a) Executing Prediction on Training data
Training Accuracy: 0.696075

Executing Prediction on Testing data
Test Accuracy: 0.618705


1b) Executing Random Prediction on Test Data
Random Accuracy: 0.197580
Accuracy Improvement over Random Prediction = 0.4211

Executing Majority Prediction on Test Data
Majority Accuracy: 0.439896
Accuracy Improvement over Majority Prediction = 0.1788


1c) Confusion Matrix is
[[15493.  3377.  1615.  1067.  2479.]
 [ 2174.  1962.   819.   266.   107.]
 [ 1124.  2935.  3646.  1137.   253.]
 [  873.  2086.  7302. 18545. 12897.]
 [  505.   478.  1149.  8343. 43086.]]
```

-Class 5 has highest diagonal entry which corresponds to the true positive ie classes identified correctly. However, if the no. of samples of a class is more, than it is possible to get a high value of that class in the confusion matrix. A better indicator is recall. -Class 1,4 and 5 have good recall values, whereas class 2 and 3 have least number of correct classification ie Model seems to classify two extreme pretty clearly.

```
1d) Executing stem/stop Prediction on Test data
Test stem/stop Accuracy: 0.606829
```

There is a reduction in accuracy on using stop word removal and stemming. The vocabulary size is also reduced.

```
1e) Executing feature engg. Prediction on Test data
Test feature engg.2 Accuracy: 0.617239
```

-Feature1: remove stemming (but have stop word removal), remove additional words that do not contribute to the review to create bigrams, and include **top 10** bigrams.
-Feature2: Increase weightage of first 3 words of the review as people tend to express more with starting words like awesome experience, worst experience ever, etc.

1f)

```
class 1 precision=0.666869 recall=0.815261 F1=0.733637
class 2 precision=0.402321 recall=0.038383 F1=0.070081
class 3 precision=0.362132 recall=0.075287 F1=0.124658
class 4 precision=0.400925 recall=0.575482 F1=0.472601
class 5 precision=0.758566 recall=0.810717 F1=0.783775
Average F1_score = 0.436950
```

A F1 macro-average will compute the metric independently for each class and then take the average (hence treating all classes equally), whereas test-error will aggregate the contributions of all classes to compute the average metric. In a multi-class classification setup, test-error is preferable if there is class imbalance (i.e have many more examples of one class than of other classes).

Q1)

```
CVXOPT package
        Linear Kernel:
                b =  1.6247072953080823
                No. of support vectors =  233
                Test Accuracy =  0.972972972972973
        Gaussian Kernel
                No. of support vectors =  1520
                b =  0.13985970004960638
                Test Accuracy =  0.9918918918918919

LIBSVM package
        Linear Kernel:
                 optimization finished, #iter = 9897
                 rho = -1.624401
                 Total nSV = 233
                 Accuracy = 97.2973% (1800/1850)

        Gaussian Kernel
                 optimization finished, #iter = 2348
                 rho = -0.140668
                 Total nSV = 1477
                 Accuracy = 99.1892% (1835/1850)
```

$$\mathbf{w} = \sum_{j=1}^{N} \alpha_j y_j \Phi(\mathbf{x}_j)$$

b = (1/n)summation over i[y(i) – w.T*phi{x(i)}]
where the summation is over support vectors and
n= No. of Support Vectors

As can be seen from test_accuracy, the accuracy of Gaussian Kernel is superior to that of linear Kernel. This is because if the data is linearly separable then linear classifier will give full accuracy. However, if data is not linearly separable, the kernel that projects data on non-linear subspace can perform better.

The computational cost for Gaussian kernel is more than that of linear kernel because it uses kernel function on each pair of data-points to project them on a higher dimensional space. The computational time of the LIBSVM is very fast compared to CVXOPT.

Training accuracy

```
Q2)
LIBSVM package
    Gaussian Kernel
         Training Accuracy = 99.92%
         Testing  Accuracy = 97.23%

LIBSVM package
    Gaussian Kernel
         Training Accuracy = 99.92% (19984/20000)
         Testing  Accuracy = 97.24% (9724/10000)
```

The result for LISVM and CVXOPT (using Gaussian Kernel and one-vs-one classifier) result in nearly same accuracy. In terms of computation time LIBSVM is much faster than CVXOPT.