Homework 3

Name: Kalpita Dapkekar

Objective function: $\min J(w) = ½\, w^2$

S.t. for all(x_i,y_i), $y_i\,(w^T Ø(x_i)) \geq 1$

a) **Using Lagrange multipliers for the constraints, write the Lagrangian objective function for this problem. What new constraints are added to this objective function?**

Using lagrange multiplier α_i

$\alpha_i\,(y_i\,(w^T Ø(x_i))-1) = 0$

Objective function: $\min L = ½\, w^2 - \sum \alpha_i(y_i(w^T Ø(x_i))-1)$

S.t $\quad \alpha_i \geq 0$

b) **Write the derivative of the the Lagrangian objective function, that you obtained in previous section with respect to w.**

$\partial L/\partial w = ½ * 2w - \sum \alpha_i\, y_i Ø(x_i)$

$= w - \sum \alpha_i\, y_i Ø(x_i)$

c) **Set the derivatives of previous section to 0 and solve for w. What do you achieve? How do you interpret these results?**

$\partial L/\partial w = 0$

Therefore,

$w = \sum \alpha_i y_i Ø(x_i)$

This is called as **weight vector**

It implies that w can be expressed as a linear combination of $Ø(x_i)$ with signed lagrange multipliers $\alpha_i y_i$ serving as a coefficient.

d) **Replace the w, obtained in part (c) in the objective function in part (b). Use your conclusions in part (c) to rewrite the objective function. The new objective function, dual Lagrangian, should be defined in terms of Lagrange multipliers only. What are the constraints that are added to the dual Lagrangian objective function?**

Objective function: $\min L = ½\, w^2 - \sum \alpha_i\,(y_i\,(w^T Ø(x_i))-1)$

$\mathbf{L}_{dual} = ½\, w^T w - \sum \alpha_i(y_i w^T Ø(x_i)) + \sum \alpha_i$

$= ½\, w^T w - w^T \sum \alpha_i(y_i Ø(x_i)) + \sum \alpha_i$

Since $w = \sum \alpha_i y_i Ø(x_i)$

$\mathbf{L}_{dual} = ½\, w^T w - w^T w + \sum \alpha_i$

$= \sum \alpha_i - \frac{1}{2} w^T w$

$\mathbf{L}_{dual} = \sum \alpha_i - \frac{1}{2} \sum\sum \alpha_i \alpha_j y_i y_j \emptyset(x_i)^T \emptyset(x_j)$

Objective function: $\mathbf{L}_{dual} = \sum \alpha_i - \frac{1}{2} \sum\sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)$

S.t $\alpha_i \geq 0$ for all x_i belongs to D

The dual Lagrangian depends only on the dot product between two vectors in feature space $\emptyset(x_i)^T \emptyset(x_j) = K(x_i, x_j)$

e) **Suppose that we want to solve the dual Lagrangian objective function, to obtain the Lagrange multipliers, using gradient ascent algorithm. Remember that gradient ascent is an iterative algorithm that updates the parameters, according to the gradient of loss with respect to the parameters, in each iteration. In other words, if the J(α) is the objective function, in iteration t + 1 we have $\alpha_{t+1} = \alpha_t + step * \partial J(\alpha)/\partial\alpha$, in which step is a constant. As a first step to do so, calculate the partial gradient of the dual Lagrangian objective function with respect to one of the Lagrange multipliers. Write the gradient in terms of the kernel function.**

Objective function: max $\partial J(\alpha) = \sum \alpha_i - \frac{1}{2} \sum\sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)$

Let the lagrange multiplier be α_k that means i=k and when i=j=k

Then $J(\alpha)$ with α_k will be

$J(\alpha_k) = \alpha_k - \frac{1}{2} \alpha_k^2 y_k^2 K(x_k, x_k) - \alpha_k y_k \sum \alpha_j y_j K(x_k, x_j)$

Partial gradient of the dual lagrange multiplier with respect to α_k , Where the Kth component of the gradient is obtained by differentiating $J(\alpha_k)$ with respect to α_k

$\partial J(\alpha_k)/\partial\alpha_k = 1 - y_k(\sum \alpha_j y_j K(x_k, x_j)) - \alpha_k y_k^2 K(x_k, x_k)$

$\alpha_{k+1} = \alpha_k +$ **step** $* (1 - y_k(\sum \alpha_j y_j K(x_k, x_j)))$

f) **Suppose that we used the result of part (e) to calculate all of the Lagrange multipliers. How can we use them to calculate w? How can we classify any new point z?**

As gradient ascent is an iterative algorithm that updates the parameter that is lagrange multiplier α_i.

So Entire **α** vector is not updated in each step, we update each component of α_k, and updated component is used to update another component of vector.

$\alpha_{t+1} = \alpha_t + step * \partial J(\alpha)/\partial\alpha$

Methods continuously updates α until $||\alpha_t - \alpha_{t+1}|| <$ epsilon

So to classify any point *z*

Y_hat = sign(h(Ø(*z*)))

= sign (w^TØ(*z*))

Since $w = \sum \alpha_i y_i \emptyset(x_i)$

$$\hat{Y} = \text{sign}\left(\sum \alpha_i y_i \varnothing(x_i)^T \varnothing(z)\right)$$
$$= \text{sign}\left(\sum \alpha_i y_i K(x_i, z)\right)$$

This way the point *z* is classified.

g) Assume that we chose a kernel function and found hard-margin kernel SVM for some data points according to some training data. Which of the following pictures can represent a hard-margin kernel SVM? Explain why, or why not, for each.

a) After using kernel function, data points are not linearly separable and there is misclassification, so not an example of Hard Margin Kernel SVM.

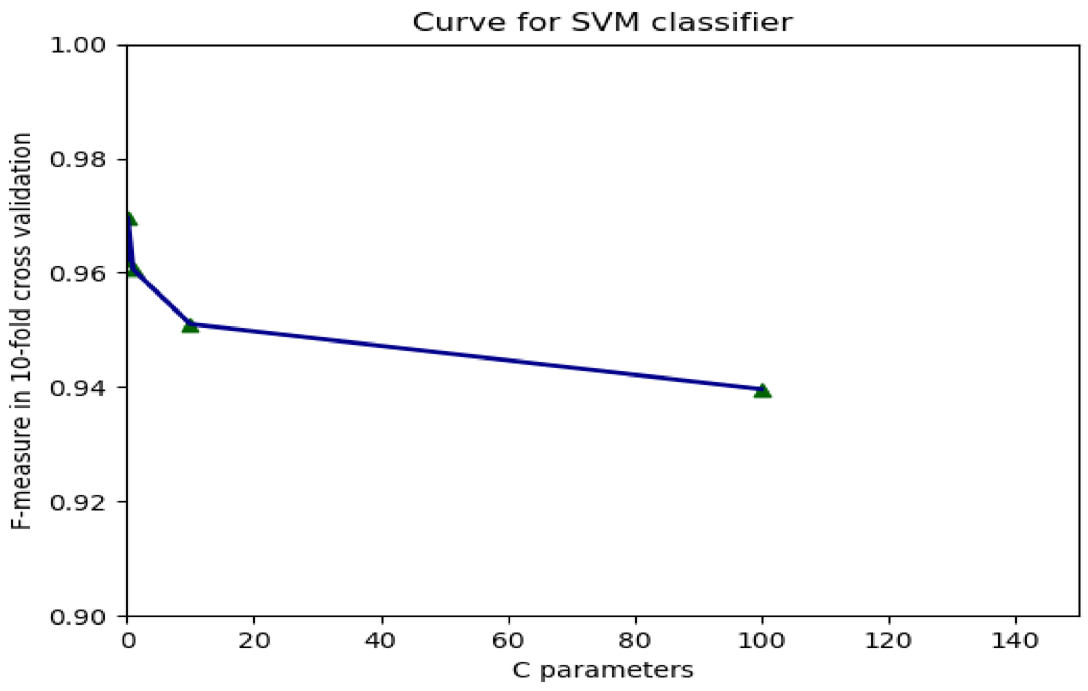
b) Datapoints are linearly separable after using kernel function no datapoints are misclassified. So **this is an example of Hard Margin Kernel SVM.**

c) Even after using Kernel function datapoints are not linearly separable. So not an example of Hard Margin Kernel SVM.

d) Even after using Kernel function datapoints are not linearly separable. So not an example of Hard Margin Kernel SVM.

Solution for Code Part.

a. C = {0.01, 0.1, 1, 10, 100}
[0.9623215169423329, 0.9696956558771388, 0.9606077933879227, 0.9510409250590064, 0.9396257362511241]



For C = 0.01
F – Measure = 0.9623215169423329

For C = 0.1
F – Measure = 0.9696956558771388

For C = 1
F – Measure = 0.9606077933879227

For C = 10
F – Measure = 0.9510409250590064

For C = 100
F – Measure = 0.9396257362511241

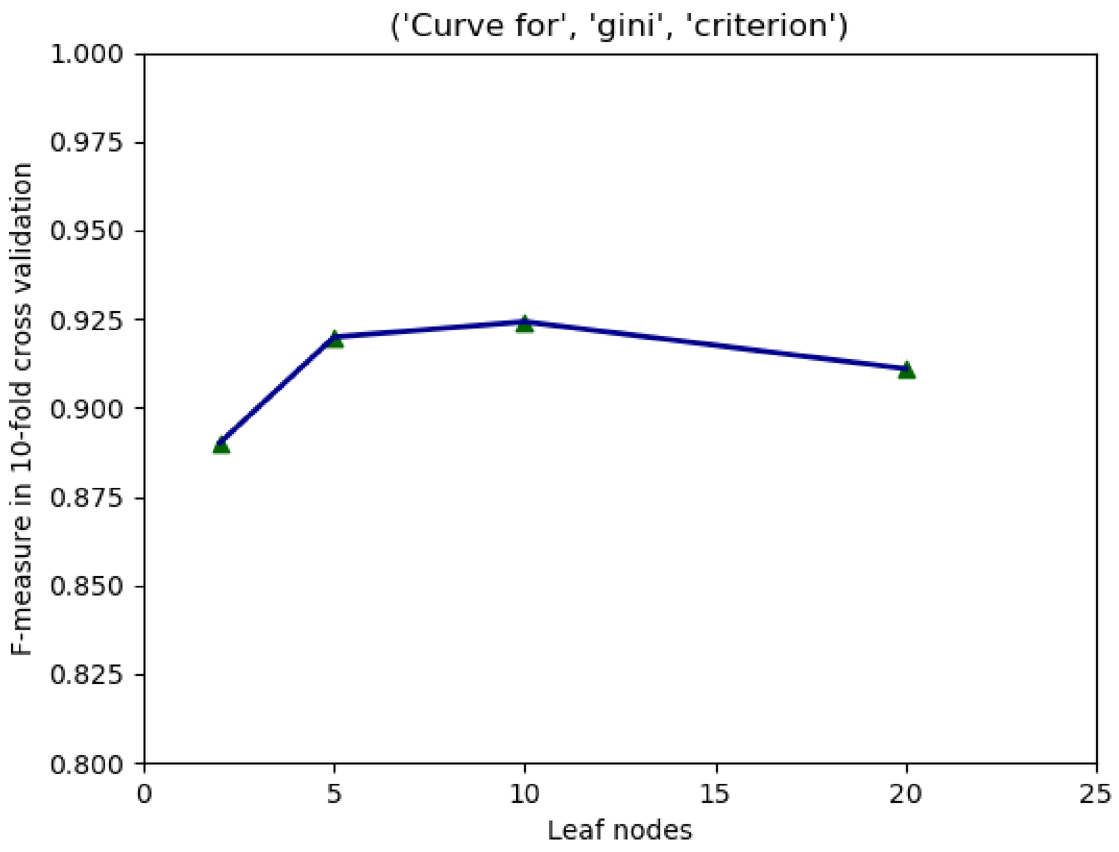
According to graph and f-measure values, C = 0.1 end up being best in cross validation. Other values are less compare to C = 0.1, there is a little difference between 0.01,0.1,1 and 10. C= 100 has the smaller F-measure value, which is not good.

b. k = {2, 5, 10, 20}
For Gini index the best leaf node is 10. The graph between the leaf node and f-measure is shown below.
K = 10 gives the best result.
[0.8901510706999037, 0.9127065461681712, 0.9186799842207032, 0.9049886141814717]
For K = 2
F – Measure = 0.8901510706999037

For K = 5
F – Measure = 0.9127065461681712

For K = 10
F – Measure = 0.9186799842207032

For K = 20
F – Measure = 0.9049886141814717



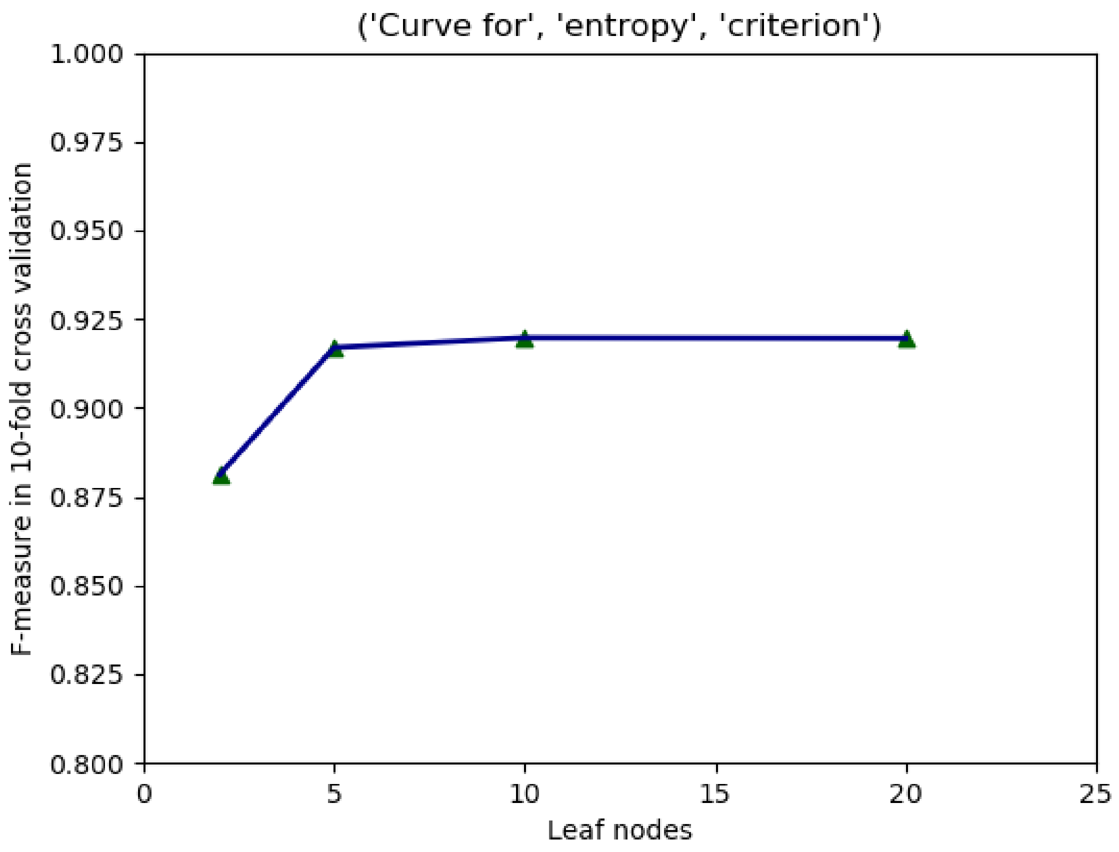
k = {2, 5, 10, 20}
For information gain the best leaf node is 20. The graph between the leaf node and f-measure is shown below.
K = 20 gives the best result.
[0.8812164995239533, 0.9169660760044703, 0.9177310935038718, 0.9246281402182339]

For K = 2
F – Measure = 0.8812164995239533

For K = 5
F – Measure = 0.9169660760044703

For K = 10
F – Measure = 0.9177310935038718

For K = 20
F – Measure = 0.9246281402182339



c. Best C = 0.1
Best k for Gini = 10
Best k for IG = 20
LDA
Random forest

Confusion Matrix for SVM

```
[[293   2]
 [  5 169]]
  precision    recall  f1-score   support

        0       0.98      0.99      0.99       295
        1       0.99      0.97      0.98       174

   micro avg       0.99      0.99      0.99       469
   macro avg       0.99      0.98      0.98       469
weighted avg       0.99      0.99      0.99       469
```

Confusion Matrix for Gini Index

```
[[291   4]
 [  4 170]]
precision    recall  f1-score   support

        0      0.99      0.99      0.99       295
        1      0.98      0.98      0.98       174

  micro avg      0.98      0.98      0.98       469
   macro avg      0.98      0.98      0.98       469
weighted avg      0.98      0.98      0.98       469
```

Confusion Matrix for Information gain

```
[[295   0]
 [  0 174]]
             precision    recall  f1-score   support

        0      1.00      1.00      1.00       295
        1      1.00      1.00      1.00       174

  micro avg      1.00      1.00      1.00       469
   macro avg      1.00      1.00      1.00       469
weighted avg      1.00      1.00      1.00       469
```

Confusion Matrix for LDA

```
[[293   2]
 [ 15 159]]
Accuracy for LDA : 0.9637526652452025
             precision    recall  f1-score   support

        0      0.95      0.99      0.97       295
        1      0.99      0.91      0.95       174

  micro avg      0.96      0.96      0.96       469
   macro avg      0.97      0.95      0.96       469
weighted avg      0.96      0.96      0.96       469
```

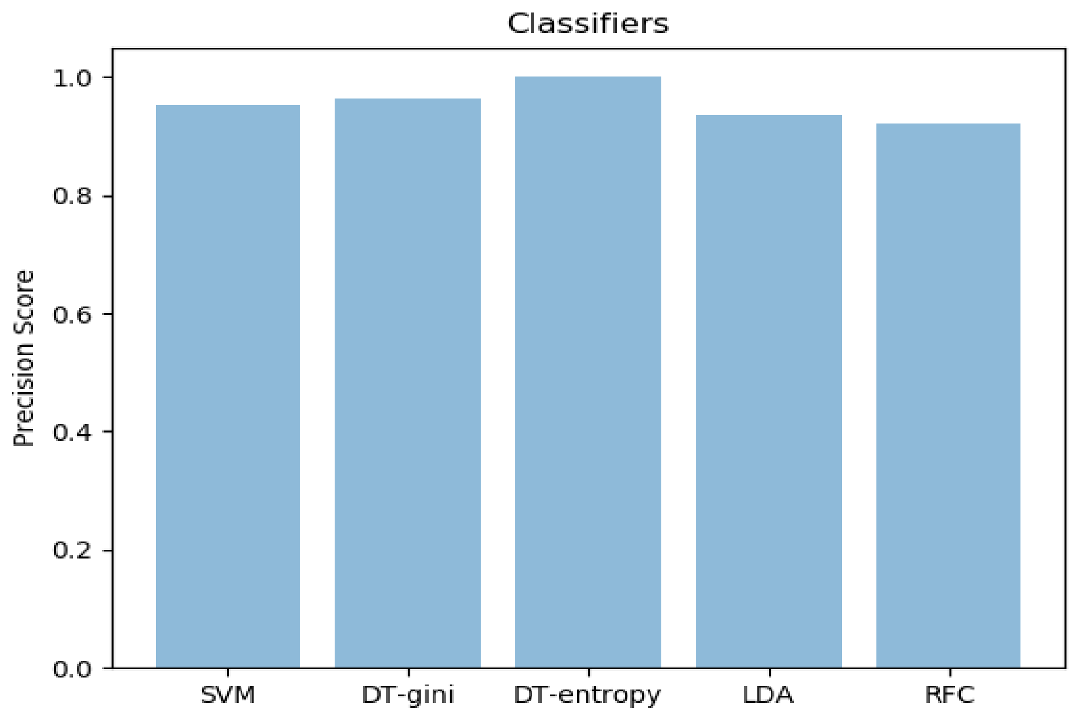
('Precision of LDA', 0.9344245787553985)

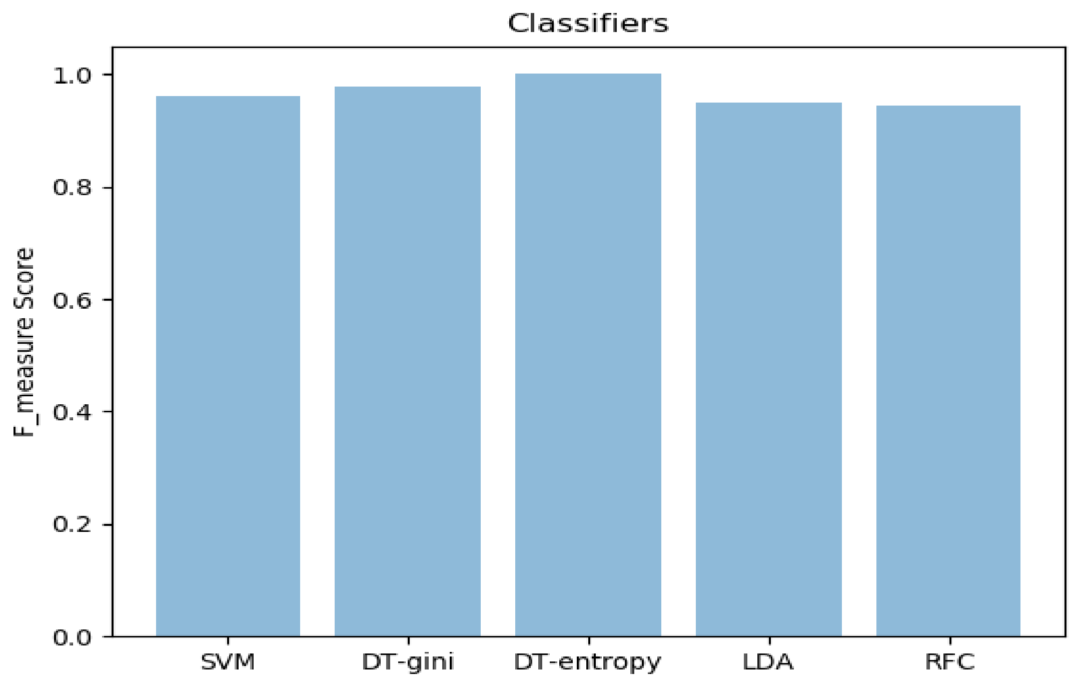
Confusion Matrix for Random Forest

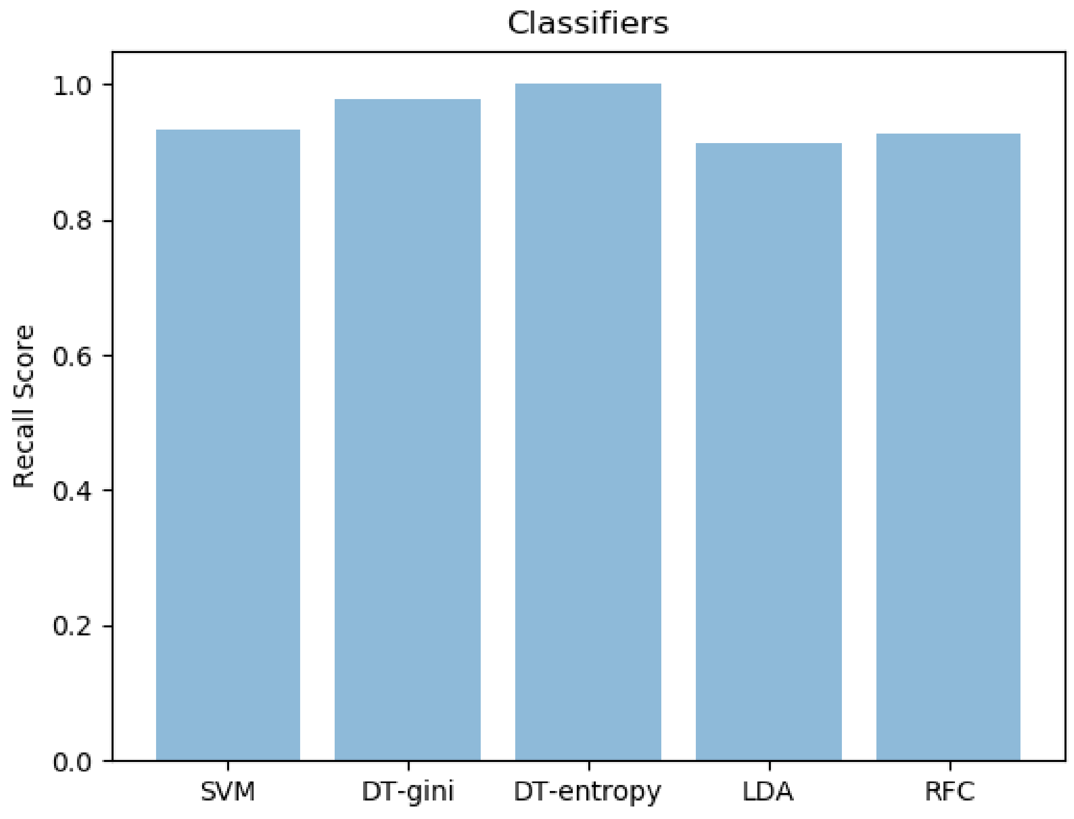
```
[[289   6]
 [ 13 161]]
```

```
Accuracy for Random forest classifier : 0.9594882729211087
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       295
           1       0.96      0.93      0.94       174

   micro avg       0.96      0.96      0.96       469
   macro avg       0.96      0.95      0.96       469
weighted avg       0.96      0.96      0.96       469
```

After plotting each classifier and metrics, Decision tree classifier using Information criterion gives best result in all the metrics. I think there is a single winner for this dataset.




Classifiers
F_measure Score
1.0
0.8
0.6
0.4
0.2
0.0
SVM
DT-gini
DT-entropy
LDA
RFC


Classifiers
Recall Score
1.0
0.8
0.6
0.4
0.2
0.0
SVM
DT-gini
DT-entropy
LDA
RFC