IoT Analytics

Project 05

TASK 1

The dataset, ‘shdeshpa.csv’, is displayed using a scatter plot as seen in Fig.1. Both scatter plots in Fig.1 are views of the same scatter plot from different angles.

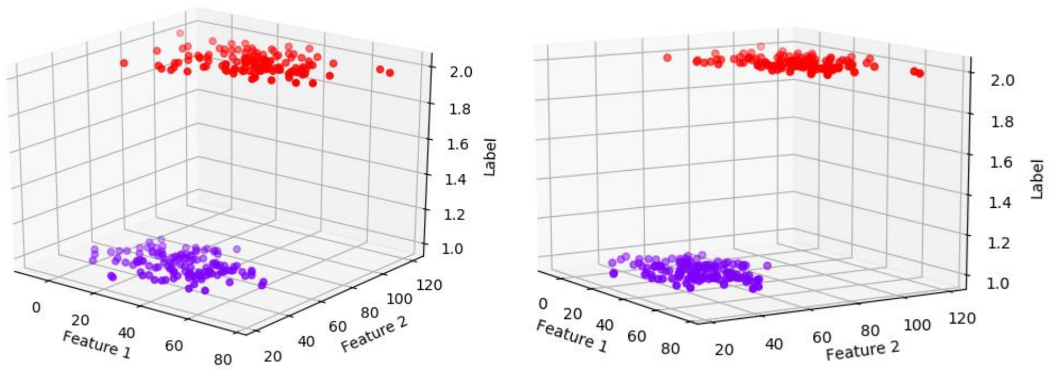
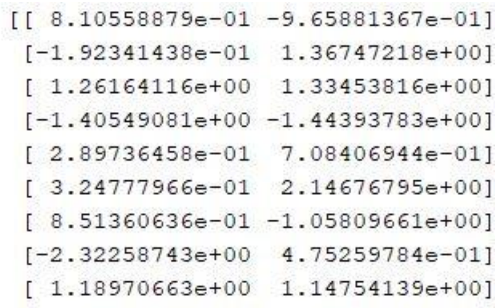


Figure 1: Scatter Plot of Dataset

From Fig.1, it can be observed that all feature data points in the dataset and each are given either label 1 or label 2 which is the case for the given dataset.

TASK 2

The values of features are scaled to values in the range [0,1].



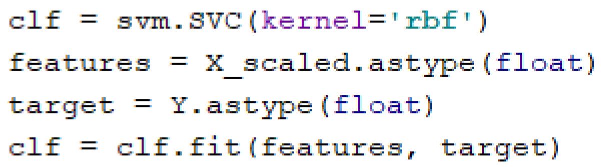
```
[[ 8.10558879e-01 -9.65881367e-01]
 [-1.92341438e-01  1.36747218e+00]
 [ 1.26164116e+00  1.33453816e+00]
 [-1.40549081e+00 -1.44393783e+00]
 [ 2.89736458e-01  7.08406944e-01]
 [ 3.24777966e-01  2.14676795e+00]
 [ 8.51360636e-01 -1.05809661e+00]
 [-2.32258743e+00  4.75259784e-01]
 [ 1.18970663e+00  1.14754139e+00]
```

Figure 2: Scaled Feature Values

In Fig.2, the first 10 scaled values of feature 1 and feature 2 of the given dataset are seen. The first value in every row is the successive scaled value of feature 1 data points and the second element value in every row is the successive scaled value of feature 2 data points.

TASK 3

The SVM method with the penalty cost is applied to the dataset and the kernel used is radial base function (RBF) (as seen in fig.3).



```
clf = svm.SVC(kernel='rbf')
features = X_scaled.astype(float)
target = Y.astype(float)
clf = clf.fit(features, target)
```

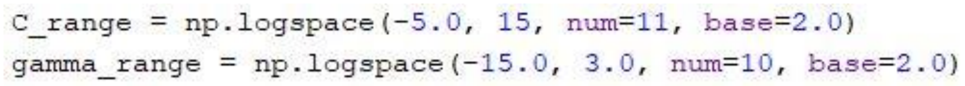
Figure 3: SVM method using RBF kernel

TASKS 4,5,6,7,8

COARSE/INITIAL SEARCH:

The RBF Kernel is applied here and is used to determine the test accuracy by using ranges of parameters C and γ.

The C range and γ range are defined using “np.logspace()” (as seen in fig.4).

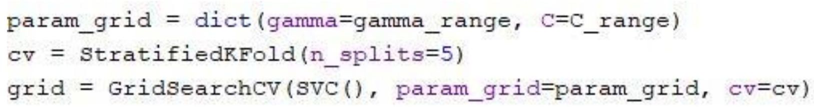


```
C_range = np.logspace(-5.0, 15, num=11, base=2.0)
gamma_range = np.logspace(-15.0, 3.0, num=10, base=2.0)
```

Figure 4: Defining C and γ range

The cross-validation of the dataset is done by splitting the dataset into 5 folds using

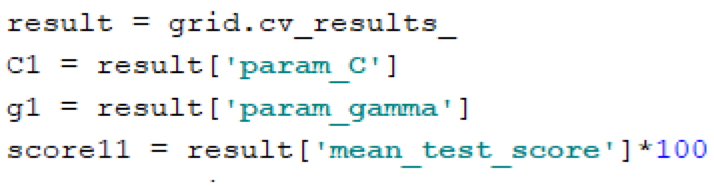
“StratifiedKFold()”. A grid search is performed using “GridSearchCV()”(Fig.5).



```
param_grid = dict(gamma=gamma_range, C=C_range)
cv = StratifiedKFold(n_splits=5)
grid = GridSearchCV(SVC(), param_grid=param_grid, cv=cv)
```

Figure 5: Code Segment for Stratified Sampling Function and Grid Search

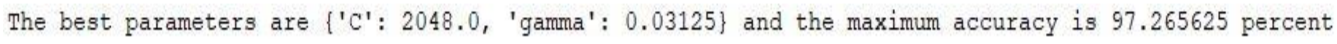
Using this range, a 3D plot is obtained using “**plot_trisurf**”. The parameters **C** and **gamma** were obtained by considering “param_C” and “param_gamma” of “**grid.cv_results_**” (Fig.6). The score considered while plotting the 3D graph is the mean test score of all the test data splits obtained at every iteration of StratifiedKFold.



```
result = grid.cv_results_
C1 = result['param_C']
g1 = result['param_gamma']
score11 = result['mean_test_score']*100
```

Figure 6: Code to obtain C and gamma parameters

The best values of C and gamma were obtained using “**grid.best_parameters_**” (Fig.7).



```
The best parameters are {'C': 2048.0, 'gamma': 0.03125} and the maximum accuracy is 97.265625 percent
```

Figure 7: Best Parameter (C, γ) Values and Maximum Accuracy for Coarse/Initial Search Range

Large C makes the cost of misclassification high ('hard margin"), thus forcing the SVM algorithm to explain the input data stricter and potentially overfit. This happens since if C is large then model chooses more data points as a support vector and higher variance and lower bias is obtained, which may lead to the problem of overfitting. This is not desirable. Hence, a finer search is performed.

Gamma defines how far the influence of a single training example reaches. If the value of Gamma is high, then the decision boundary will depend on points close to the decision boundary and nearer points will carry more weight than far away points due to which the decision boundary becomes wigglier. If the value of Gamma is low, then far away points carry more weights than nearer points. As a result of this, the decision boundary becomes more like a straight line.

In the coarse search, the value of ‘C’ is high, and this might lead to the problem of overfitting as cost of misclassification is high. Value of Gamma is neither too high nor too low, and hence, can be considered an optimum parameter value.

The 3D plot obtained is shown in Fig.8.

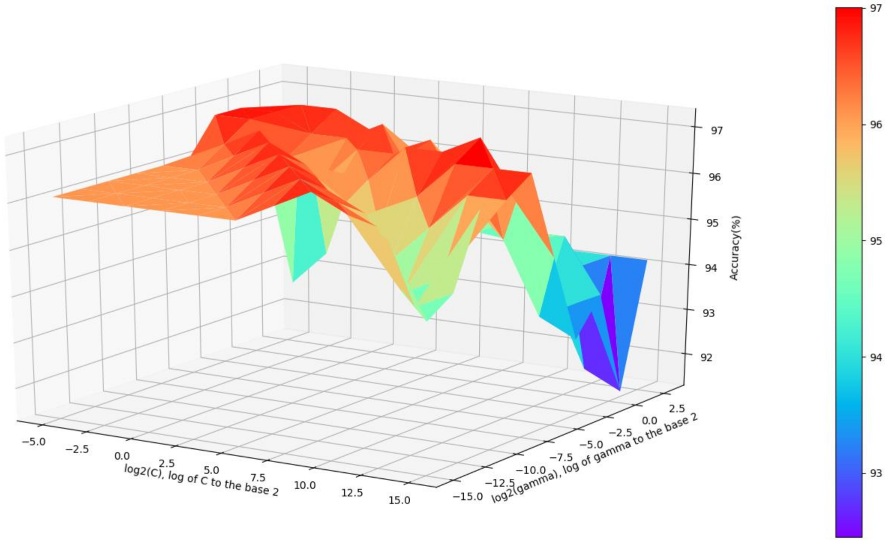
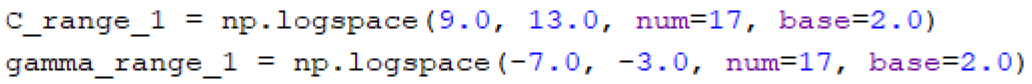


Figure 8: 3D plot of percent Accuracy vs. parameters

It is observed that the best (C, γ) is ($2^{11}, 2^{-5}$) with the accuracy being 97.266 % (Fig.7, Fig.8).

FINER SEARCH:

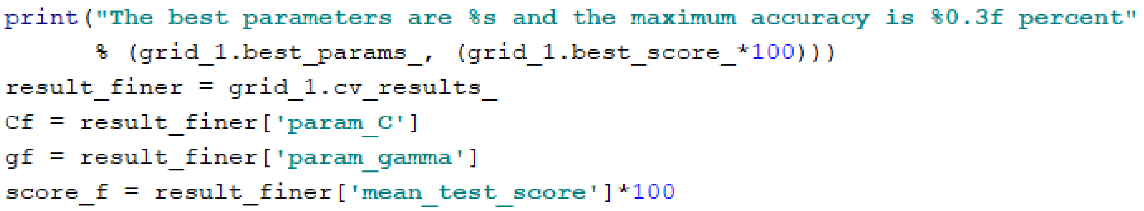
From the results obtained above, a finer search is conducted on the neighborhood of ($2^{11}, 2^{-5}$). The range of values of C, γ used for finer search are seen in Fig.9.



```
C_range_1 = np.logspace(9.0, 13.0, num=17, base=2.0)
gamma_range_1 = np.logspace(-7.0, -3.0, num=17, base=2.0)
```

Figure 9: Range of Values of C and gamma for finer search

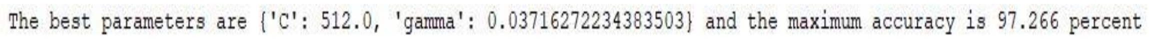
The values of C and gamma, and their best values are obtained using the code snippet shown in Fig.10.



```
print("The best parameters are %s and the maximum accuracy is %0.3f percent"
      % (grid_1.best_params_, (grid_1.best_score_*100)))
result_finer = grid_1.cv_results_
Cf = result_finer['param_C']
gf = result_finer['param_gamma']
score_f = result_finer['mean_test_score']*100
```

Figure 10: Code Snippet for obtaining values of C and gamma, and, best values of C and gamma

After performing a finer search, the values for C, γ, change. However, the accuracy remains the same. This can be seen in Fig.11.



```
The best parameters are {'C': 512.0, 'gamma': 0.03716272234383503} and the maximum accuracy is 97.266 percent
```

Figure 11: Best Parameter (C, γ) Values and Maximum Accuracy for Finer Search Range

The 3D Plot obtained for finer search is seen in Fig.12.

It is observed that the best value of (C, γ) is $(2^9, 2^{-4.75})$ with the accuracy 97.266 % (Fig.11, Fig.12).

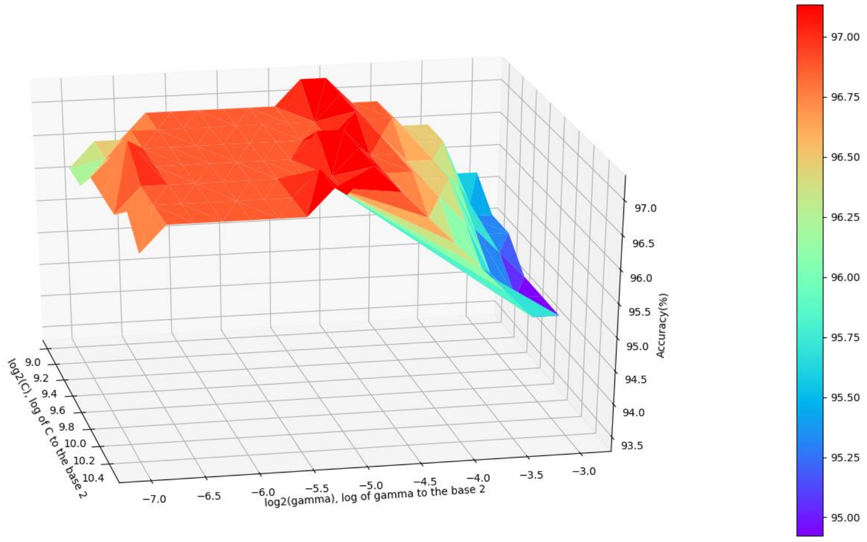


Figure 12: 3D Plot for Finer Search

The best values for C and gamma observed for finer search are:

C = 512.0

gamma = 0.0371

The best value of C for a finer search is lower in comparison to the best value of C obtained using coarse search. Hence, the chances of overfitting due to a large C value are mitigated and the value obtained for through finer search C can be considered optimum.

The value of gamma obtained using finer search is neither very high nor too low. Hence, the gamma value is considered to be optimum for the considered range.