2020_1125_SVM_Classification

November 27, 2020

1 Classifying Pulsars from the High Time Resolution Universe Survey (HTRU2) - Support Vector Machine (SVM) Classification

1.1 Overview & Citation

In this code notebook, we attempt to classify pulsars from the High Time Resolution Universe Survey, South (HTRU2) dataset using support vector machine (SVM) classification. The dataset was retrieved from the UC Irvine Machine Learning Repository at the following link: https://archive.ics.uci.edu/ml/datasets/HTRU2#.

The dataset was donated to the UCI Repository by Dr. Robert Lyon of The University of Manchester, United Kingdom. The two papers requested for citation in the description are listed below:

- R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, J. D. Knowles, Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach, Monthly Notices of the Royal Astronomical Society 459 (1), 1104-1123, DOI: 10.1093/mnras/stw656
- R. J. Lyon, HTRU2, DOI: 10.6084/m9.figshare.3080389.v1.

1.2 Import the Relevant Libraries

[12]:
```
# Data Manipulation
import pandas as pd
import numpy as np

# Modeling & Evaluation
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,classification_report
```

1.3 Import & Check the Data

[2]:
```
df = pd.read_csv('2020_1125_Pulsar_Data.csv')
pulsar_data = df.copy()
```

[3]:
```
pulsar_data.head()
```

[3]:
```
       IP_Mean  IP_StdDev  IP_Kurtosis  IP_Skewness   DM_Mean  DM_StdDev  \
0  140.562500  55.683782    -0.234571    -0.699648  3.199833  19.110426
1  102.507812  58.882430     0.465318    -0.515088  1.677258  14.860146
2  103.015625  39.341649     0.323328     1.051164  3.121237  21.744669
3  136.750000  57.178449    -0.068415    -0.636238  3.642977  20.959280
4   88.726562  40.672225     0.600866     1.123492  1.178930  11.468720

   DM_Kurtosis  DM_Skewness  Class
0     7.975532    74.242225      0
1    10.576487   127.393580      0
2     7.735822    63.171909      0
3     6.896499    53.593661      0
4    14.269573   252.567306      0
```

1.4 Train Test Split

[4]:
```
X = pulsar_data.drop('Class',axis=1)
y = pulsar_data['Class']
```

[5]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
↪random_state=42)
```

1.5 SVM with Default Parameters

1.5.1 Construct and Test the Model

[6]:
```
svm_model = SVC()
svm_model.fit(X_train,y_train)
```

[6]:
```
SVC()
```

[7]:
```
y_pred = svm_model.predict(X_test)
```

1.5.2 Model Evaluation

[8]:
```
confusion = confusion_matrix(y_test,y_pred)
print(f'CONFUSION MATRIX:
↪\n\n{confusion[0][0]}\t{confusion[0][1]}\n{confusion[1][0]}\t{confusion[1][1]}')
```

```
CONFUSION MATRIX:

4050    20
96      309
```

[9]:
```
print(f'CLASSIFICATION REPORT:\n\n{classification_report(y_test,y_pred)}')
```

```
CLASSIFICATION REPORT:

              precision    recall  f1-score   support

           0       0.98      1.00      0.99      4070
           1       0.94      0.76      0.84       405

    accuracy                           0.97      4475
   macro avg       0.96      0.88      0.91      4475
weighted avg       0.97      0.97      0.97      4475
```

The dataset contains a total of 1,639 actual pulsars out of 16,259 instances in the dataset (approximately 10%). This means that we have an unbalanced classification problem, and accuracy is not a good metric. Therefore, the most important metrics for predicting a pulsar with this model are: * Precision = 0.94 * Recall = 0.76 * F1-Score = 0.84

1.6 Optimizing Performance

1.6.1 Run a Cross-Validation Grid Search

Let's try to improve the evaluation metrics by running a grid search to optimize the C and gamma parameters of the support vector classifier below:

[13]:
```
grid_parameters = {'C': [0.1,1, 10, 100, 1000], 'gamma': [1,0.1,0.01,0.001,0.
↪0001], 'kernel': ['rbf']}
```

[16]:
```
grid = GridSearchCV(SVC(),grid_parameters,refit=True,verbose=3)
```

[17]:
```
# Note - this will take a few minutes to run
grid.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV] C=0.1, gamma=1, kernel=rbf …

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.

[CV] … C=0.1, gamma=1, kernel=rbf, score=0.908, total=   5.3s
[CV] C=0.1, gamma=1, kernel=rbf …

[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    5.3s remaining:    0.0s

[CV] … C=0.1, gamma=1, kernel=rbf, score=0.908, total=   5.0s
[CV] C=0.1, gamma=1, kernel=rbf …

[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:   10.3s remaining:    0.0s

[CV] … C=0.1, gamma=1, kernel=rbf, score=0.908, total=   5.1s
[CV] C=0.1, gamma=1, kernel=rbf …
[CV] … C=0.1, gamma=1, kernel=rbf, score=0.908, total=   5.1s
[CV] C=0.1, gamma=1, kernel=rbf …
[CV] … C=0.1, gamma=1, kernel=rbf, score=0.908, total=   5.1s
```

```
[CV] C=0.1, gamma=0.1, kernel=rbf …
[CV] … C=0.1, gamma=0.1, kernel=rbf, score=0.908, total=   4.3s
[CV] C=0.1, gamma=0.1, kernel=rbf …
[CV] … C=0.1, gamma=0.1, kernel=rbf, score=0.908, total=   4.3s
[CV] C=0.1, gamma=0.1, kernel=rbf …
[CV] … C=0.1, gamma=0.1, kernel=rbf, score=0.908, total=   4.6s
[CV] C=0.1, gamma=0.1, kernel=rbf …
[CV] … C=0.1, gamma=0.1, kernel=rbf, score=0.908, total=   4.4s
[CV] C=0.1, gamma=0.1, kernel=rbf …
[CV] … C=0.1, gamma=0.1, kernel=rbf, score=0.908, total=   4.3s
[CV] C=0.1, gamma=0.01, kernel=rbf …
[CV] … C=0.1, gamma=0.01, kernel=rbf, score=0.932, total=   1.3s
[CV] C=0.1, gamma=0.01, kernel=rbf …
[CV] … C=0.1, gamma=0.01, kernel=rbf, score=0.934, total=   1.5s
[CV] C=0.1, gamma=0.01, kernel=rbf …
[CV] … C=0.1, gamma=0.01, kernel=rbf, score=0.929, total=   1.3s
[CV] C=0.1, gamma=0.01, kernel=rbf …
[CV] … C=0.1, gamma=0.01, kernel=rbf, score=0.934, total=   1.4s
[CV] C=0.1, gamma=0.01, kernel=rbf …
[CV] … C=0.1, gamma=0.01, kernel=rbf, score=0.928, total=   1.4s
[CV] C=0.1, gamma=0.001, kernel=rbf …
[CV] … C=0.1, gamma=0.001, kernel=rbf, score=0.968, total=   0.4s
[CV] C=0.1, gamma=0.001, kernel=rbf …
[CV] … C=0.1, gamma=0.001, kernel=rbf, score=0.974, total=   0.4s
[CV] C=0.1, gamma=0.001, kernel=rbf …
[CV] … C=0.1, gamma=0.001, kernel=rbf, score=0.973, total=   0.4s
[CV] C=0.1, gamma=0.001, kernel=rbf …
[CV] … C=0.1, gamma=0.001, kernel=rbf, score=0.973, total=   0.4s
[CV] C=0.1, gamma=0.001, kernel=rbf …
[CV] … C=0.1, gamma=0.001, kernel=rbf, score=0.973, total=   0.4s
[CV] C=0.1, gamma=0.0001, kernel=rbf …
[CV] … C=0.1, gamma=0.0001, kernel=rbf, score=0.967, total=   0.3s
[CV] C=0.1, gamma=0.0001, kernel=rbf …
[CV] … C=0.1, gamma=0.0001, kernel=rbf, score=0.975, total=   0.3s
[CV] C=0.1, gamma=0.0001, kernel=rbf …
[CV] … C=0.1, gamma=0.0001, kernel=rbf, score=0.972, total=   0.3s
[CV] C=0.1, gamma=0.0001, kernel=rbf …
[CV] … C=0.1, gamma=0.0001, kernel=rbf, score=0.973, total=   0.3s
[CV] C=0.1, gamma=0.0001, kernel=rbf …
[CV] … C=0.1, gamma=0.0001, kernel=rbf, score=0.969, total=   0.3s
[CV] C=1, gamma=1, kernel=rbf …
[CV] … C=1, gamma=1, kernel=rbf, score=0.908, total=   6.4s
[CV] C=1, gamma=1, kernel=rbf …
[CV] … C=1, gamma=1, kernel=rbf, score=0.908, total=   6.3s
[CV] C=1, gamma=1, kernel=rbf …
[CV] … C=1, gamma=1, kernel=rbf, score=0.908, total=   6.2s
[CV] C=1, gamma=1, kernel=rbf …
[CV] … C=1, gamma=1, kernel=rbf, score=0.908, total=   6.1s
```

```
[CV] C=1, gamma=1, kernel=rbf …
[CV] … C=1, gamma=1, kernel=rbf, score=0.908, total=   6.1s
[CV] C=1, gamma=0.1, kernel=rbf …
[CV] … C=1, gamma=0.1, kernel=rbf, score=0.911, total=   5.8s
[CV] C=1, gamma=0.1, kernel=rbf …
[CV] … C=1, gamma=0.1, kernel=rbf, score=0.911, total=   5.8s
[CV] C=1, gamma=0.1, kernel=rbf …
[CV] … C=1, gamma=0.1, kernel=rbf, score=0.911, total=   5.9s
[CV] C=1, gamma=0.1, kernel=rbf …
[CV] … C=1, gamma=0.1, kernel=rbf, score=0.911, total=   5.8s
[CV] C=1, gamma=0.1, kernel=rbf …
[CV] … C=1, gamma=0.1, kernel=rbf, score=0.911, total=   5.8s
[CV] C=1, gamma=0.01, kernel=rbf …
[CV] … C=1, gamma=0.01, kernel=rbf, score=0.968, total=   1.2s
[CV] C=1, gamma=0.01, kernel=rbf …
[CV] … C=1, gamma=0.01, kernel=rbf, score=0.971, total=   1.3s
[CV] C=1, gamma=0.01, kernel=rbf …
[CV] … C=1, gamma=0.01, kernel=rbf, score=0.970, total=   1.3s
[CV] C=1, gamma=0.01, kernel=rbf …
[CV] … C=1, gamma=0.01, kernel=rbf, score=0.969, total=   1.3s
[CV] C=1, gamma=0.01, kernel=rbf …
[CV] … C=1, gamma=0.01, kernel=rbf, score=0.969, total=   1.3s
[CV] C=1, gamma=0.001, kernel=rbf …
[CV] … C=1, gamma=0.001, kernel=rbf, score=0.971, total=   0.4s
[CV] C=1, gamma=0.001, kernel=rbf …
[CV] … C=1, gamma=0.001, kernel=rbf, score=0.977, total=   0.4s
[CV] C=1, gamma=0.001, kernel=rbf …
[CV] … C=1, gamma=0.001, kernel=rbf, score=0.978, total=   0.4s
[CV] C=1, gamma=0.001, kernel=rbf …
[CV] … C=1, gamma=0.001, kernel=rbf, score=0.974, total=   0.4s
[CV] C=1, gamma=0.001, kernel=rbf …
[CV] … C=1, gamma=0.001, kernel=rbf, score=0.974, total=   0.4s
[CV] C=1, gamma=0.0001, kernel=rbf …
[CV] … C=1, gamma=0.0001, kernel=rbf, score=0.967, total=   0.3s
[CV] C=1, gamma=0.0001, kernel=rbf …
[CV] … C=1, gamma=0.0001, kernel=rbf, score=0.977, total=   0.3s
[CV] C=1, gamma=0.0001, kernel=rbf …
[CV] … C=1, gamma=0.0001, kernel=rbf, score=0.974, total=   0.3s
[CV] C=1, gamma=0.0001, kernel=rbf …
[CV] … C=1, gamma=0.0001, kernel=rbf, score=0.973, total=   0.3s
[CV] C=1, gamma=0.0001, kernel=rbf …
[CV] … C=1, gamma=0.0001, kernel=rbf, score=0.971, total=   0.3s
[CV] C=10, gamma=1, kernel=rbf …
[CV] … C=10, gamma=1, kernel=rbf, score=0.908, total=   5.2s
[CV] C=10, gamma=1, kernel=rbf …
[CV] … C=10, gamma=1, kernel=rbf, score=0.908, total=   5.4s
[CV] C=10, gamma=1, kernel=rbf …
[CV] … C=10, gamma=1, kernel=rbf, score=0.908, total=   5.3s
```

```
[CV] C=10, gamma=1, kernel=rbf …
[CV] … C=10, gamma=1, kernel=rbf, score=0.908, total=   5.4s
[CV] C=10, gamma=1, kernel=rbf …
[CV] … C=10, gamma=1, kernel=rbf, score=0.908, total=   6.1s
[CV] C=10, gamma=0.1, kernel=rbf …
[CV] … C=10, gamma=0.1, kernel=rbf, score=0.911, total=   7.1s
[CV] C=10, gamma=0.1, kernel=rbf …
[CV] … C=10, gamma=0.1, kernel=rbf, score=0.913, total=   7.1s
[CV] C=10, gamma=0.1, kernel=rbf …
[CV] … C=10, gamma=0.1, kernel=rbf, score=0.911, total=   6.9s
[CV] C=10, gamma=0.1, kernel=rbf …
[CV] … C=10, gamma=0.1, kernel=rbf, score=0.911, total=   7.2s
[CV] C=10, gamma=0.1, kernel=rbf …
[CV] … C=10, gamma=0.1, kernel=rbf, score=0.910, total=   7.0s
[CV] C=10, gamma=0.01, kernel=rbf …
[CV] … C=10, gamma=0.01, kernel=rbf, score=0.971, total=   1.2s
[CV] C=10, gamma=0.01, kernel=rbf …
[CV] … C=10, gamma=0.01, kernel=rbf, score=0.972, total=   1.1s
[CV] C=10, gamma=0.01, kernel=rbf …
[CV] … C=10, gamma=0.01, kernel=rbf, score=0.971, total=   1.1s
[CV] C=10, gamma=0.01, kernel=rbf …
[CV] … C=10, gamma=0.01, kernel=rbf, score=0.970, total=   1.1s
[CV] C=10, gamma=0.01, kernel=rbf …
[CV] … C=10, gamma=0.01, kernel=rbf, score=0.970, total=   1.1s
[CV] C=10, gamma=0.001, kernel=rbf …
[CV] … C=10, gamma=0.001, kernel=rbf, score=0.976, total=   0.5s
[CV] C=10, gamma=0.001, kernel=rbf …
[CV] … C=10, gamma=0.001, kernel=rbf, score=0.979, total=   0.5s
[CV] C=10, gamma=0.001, kernel=rbf …
[CV] … C=10, gamma=0.001, kernel=rbf, score=0.978, total=   0.5s
[CV] C=10, gamma=0.001, kernel=rbf …
[CV] … C=10, gamma=0.001, kernel=rbf, score=0.978, total=   0.5s
[CV] C=10, gamma=0.001, kernel=rbf …
[CV] … C=10, gamma=0.001, kernel=rbf, score=0.977, total=   0.5s
[CV] C=10, gamma=0.0001, kernel=rbf …
[CV] … C=10, gamma=0.0001, kernel=rbf, score=0.972, total=   0.3s
[CV] C=10, gamma=0.0001, kernel=rbf …
[CV] … C=10, gamma=0.0001, kernel=rbf, score=0.977, total=   0.3s
[CV] C=10, gamma=0.0001, kernel=rbf …
[CV] … C=10, gamma=0.0001, kernel=rbf, score=0.977, total=   0.3s
[CV] C=10, gamma=0.0001, kernel=rbf …
[CV] … C=10, gamma=0.0001, kernel=rbf, score=0.976, total=   0.3s
[CV] C=10, gamma=0.0001, kernel=rbf …
[CV] … C=10, gamma=0.0001, kernel=rbf, score=0.974, total=   0.3s
[CV] C=100, gamma=1, kernel=rbf …
[CV] … C=100, gamma=1, kernel=rbf, score=0.908, total=   5.5s
[CV] C=100, gamma=1, kernel=rbf …
[CV] … C=100, gamma=1, kernel=rbf, score=0.908, total=   5.6s
```

```
[CV] C=100, gamma=1, kernel=rbf …
[CV] … C=100, gamma=1, kernel=rbf, score=0.908, total=   5.6s
[CV] C=100, gamma=1, kernel=rbf …
[CV] … C=100, gamma=1, kernel=rbf, score=0.908, total=   5.5s
[CV] C=100, gamma=1, kernel=rbf …
[CV] … C=100, gamma=1, kernel=rbf, score=0.908, total=   5.7s
[CV] C=100, gamma=0.1, kernel=rbf …
[CV] … C=100, gamma=0.1, kernel=rbf, score=0.911, total=   6.9s
[CV] C=100, gamma=0.1, kernel=rbf …
[CV] … C=100, gamma=0.1, kernel=rbf, score=0.913, total=   6.8s
[CV] C=100, gamma=0.1, kernel=rbf …
[CV] … C=100, gamma=0.1, kernel=rbf, score=0.911, total=   7.0s
[CV] C=100, gamma=0.1, kernel=rbf …
[CV] … C=100, gamma=0.1, kernel=rbf, score=0.911, total=   7.1s
[CV] C=100, gamma=0.1, kernel=rbf …
[CV] … C=100, gamma=0.1, kernel=rbf, score=0.910, total=   8.2s
[CV] C=100, gamma=0.01, kernel=rbf …
[CV] … C=100, gamma=0.01, kernel=rbf, score=0.960, total=   1.1s
[CV] C=100, gamma=0.01, kernel=rbf …
[CV] … C=100, gamma=0.01, kernel=rbf, score=0.963, total=   1.0s
[CV] C=100, gamma=0.01, kernel=rbf …
[CV] … C=100, gamma=0.01, kernel=rbf, score=0.962, total=   1.0s
[CV] C=100, gamma=0.01, kernel=rbf …
[CV] … C=100, gamma=0.01, kernel=rbf, score=0.960, total=   1.1s
[CV] C=100, gamma=0.01, kernel=rbf …
[CV] … C=100, gamma=0.01, kernel=rbf, score=0.961, total=   1.1s
[CV] C=100, gamma=0.001, kernel=rbf …
[CV] … C=100, gamma=0.001, kernel=rbf, score=0.974, total=   0.8s
[CV] C=100, gamma=0.001, kernel=rbf …
[CV] … C=100, gamma=0.001, kernel=rbf, score=0.978, total=   0.8s
[CV] C=100, gamma=0.001, kernel=rbf …
[CV] … C=100, gamma=0.001, kernel=rbf, score=0.977, total=   0.8s
[CV] C=100, gamma=0.001, kernel=rbf …
[CV] … C=100, gamma=0.001, kernel=rbf, score=0.976, total=   0.8s
[CV] C=100, gamma=0.001, kernel=rbf …
[CV] … C=100, gamma=0.001, kernel=rbf, score=0.975, total=   0.8s
[CV] C=100, gamma=0.0001, kernel=rbf …
[CV] … C=100, gamma=0.0001, kernel=rbf, score=0.977, total=   0.4s
[CV] C=100, gamma=0.0001, kernel=rbf …
[CV] … C=100, gamma=0.0001, kernel=rbf, score=0.980, total=   0.4s
[CV] C=100, gamma=0.0001, kernel=rbf …
[CV] … C=100, gamma=0.0001, kernel=rbf, score=0.980, total=   0.5s
[CV] C=100, gamma=0.0001, kernel=rbf …
[CV] … C=100, gamma=0.0001, kernel=rbf, score=0.980, total=   0.5s
[CV] C=100, gamma=0.0001, kernel=rbf …
[CV] … C=100, gamma=0.0001, kernel=rbf, score=0.978, total=   0.4s
[CV] C=1000, gamma=1, kernel=rbf …
[CV] … C=1000, gamma=1, kernel=rbf, score=0.908, total=   5.4s
```

```
[CV] C=1000, gamma=1, kernel=rbf …
[CV] … C=1000, gamma=1, kernel=rbf, score=0.908, total=   5.8s
[CV] C=1000, gamma=1, kernel=rbf …
[CV] … C=1000, gamma=1, kernel=rbf, score=0.908, total=   5.7s
[CV] C=1000, gamma=1, kernel=rbf …
[CV] … C=1000, gamma=1, kernel=rbf, score=0.908, total=   5.5s
[CV] C=1000, gamma=1, kernel=rbf …
[CV] … C=1000, gamma=1, kernel=rbf, score=0.908, total=   6.1s
[CV] C=1000, gamma=0.1, kernel=rbf …
[CV] … C=1000, gamma=0.1, kernel=rbf, score=0.911, total=   7.4s
[CV] C=1000, gamma=0.1, kernel=rbf …
[CV] … C=1000, gamma=0.1, kernel=rbf, score=0.913, total=   7.1s
[CV] C=1000, gamma=0.1, kernel=rbf …
[CV] … C=1000, gamma=0.1, kernel=rbf, score=0.911, total=   7.4s
[CV] C=1000, gamma=0.1, kernel=rbf …
[CV] … C=1000, gamma=0.1, kernel=rbf, score=0.911, total=   7.4s
[CV] C=1000, gamma=0.1, kernel=rbf …
[CV] … C=1000, gamma=0.1, kernel=rbf, score=0.910, total=   7.0s
[CV] C=1000, gamma=0.01, kernel=rbf …
[CV] … C=1000, gamma=0.01, kernel=rbf, score=0.954, total=   1.1s
[CV] C=1000, gamma=0.01, kernel=rbf …
[CV] … C=1000, gamma=0.01, kernel=rbf, score=0.955, total=   0.9s
[CV] C=1000, gamma=0.01, kernel=rbf …
[CV] … C=1000, gamma=0.01, kernel=rbf, score=0.953, total=   0.9s
[CV] C=1000, gamma=0.01, kernel=rbf …
[CV] … C=1000, gamma=0.01, kernel=rbf, score=0.956, total=   0.9s
[CV] C=1000, gamma=0.01, kernel=rbf …
[CV] … C=1000, gamma=0.01, kernel=rbf, score=0.948, total=   1.1s
[CV] C=1000, gamma=0.001, kernel=rbf …
[CV] … C=1000, gamma=0.001, kernel=rbf, score=0.974, total=   2.2s
[CV] C=1000, gamma=0.001, kernel=rbf …
[CV] … C=1000, gamma=0.001, kernel=rbf, score=0.978, total=   2.7s
[CV] C=1000, gamma=0.001, kernel=rbf …
[CV] … C=1000, gamma=0.001, kernel=rbf, score=0.977, total=   2.6s
[CV] C=1000, gamma=0.001, kernel=rbf …
[CV] … C=1000, gamma=0.001, kernel=rbf, score=0.975, total=   2.5s
[CV] C=1000, gamma=0.001, kernel=rbf …
[CV] … C=1000, gamma=0.001, kernel=rbf, score=0.977, total=   2.4s
[CV] C=1000, gamma=0.0001, kernel=rbf …
[CV] … C=1000, gamma=0.0001, kernel=rbf, score=0.978, total=   1.2s
[CV] C=1000, gamma=0.0001, kernel=rbf …
[CV] … C=1000, gamma=0.0001, kernel=rbf, score=0.983, total=   1.3s
[CV] C=1000, gamma=0.0001, kernel=rbf …
[CV] … C=1000, gamma=0.0001, kernel=rbf, score=0.979, total=   1.2s
[CV] C=1000, gamma=0.0001, kernel=rbf …
[CV] … C=1000, gamma=0.0001, kernel=rbf, score=0.981, total=   1.5s
[CV] C=1000, gamma=0.0001, kernel=rbf …
[CV] … C=1000, gamma=0.0001, kernel=rbf, score=0.978, total=   1.5s
```

```
[Parallel(n_jobs=1)]: Done 125 out of 125 | elapsed:  6.1min finished
```

[17]:
```
GridSearchCV(estimator=SVC(),
             param_grid={'C': [0.1, 1, 10, 100, 1000],
                         'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                         'kernel': ['rbf']},
             verbose=3)
```

[18]:
```
# Find the best parameters:
grid.best_params_
```

[18]:
```
{'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'}
```

[19]:
```
# Find the best estimator:
grid.best_estimator_
```

[19]:
```
SVC(C=1000, gamma=0.0001)
```

1.6.2 Run the Optimized Model

[21]:
```
svm_optimized = SVC(C=1000, gamma=0.0001)
svm_optimized.fit(X_train,y_train)
y_pred_optimized = svm_optimized.predict(X_test)
```

1.6.3 Model Evaluation

[23]:
```
confusion = confusion_matrix(y_test,y_pred_optimized)
print(f'CONFUSION MATRIX:
 ↪\n\n{confusion[0][0]}\t{confusion[0][1]}\n{confusion[1][0]}\t{confusion[1][1]}')
```

```
CONFUSION MATRIX:

4047    23
68      337
```

[24]:
```
print(f'CLASSIFICATION REPORT:
 ↪\n\n{classification_report(y_test,y_pred_optimized)}')
```

```
CLASSIFICATION REPORT:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.93      4475
```

```
weighted avg       0.98      0.98      0.98      4475
```

1.7 Conclusions

Running the cross-validation grid search resulted in the following improvements: * Accuracy increased from 0.97 to 0.98 * Recall increased from 0.76 to 0.83 * F1-Score increased from 0.84 to 0.88

Let's save the results of the optimized model for future reference:

[25]:
```
with open("2020_1125_SVM_Results.csv","w") as file:
    file.write('Model,Accuracy,Precision,Recall,F1-Score\n')
    file.write('SVM,0.98,0.94,0.83,0.88\n')
```