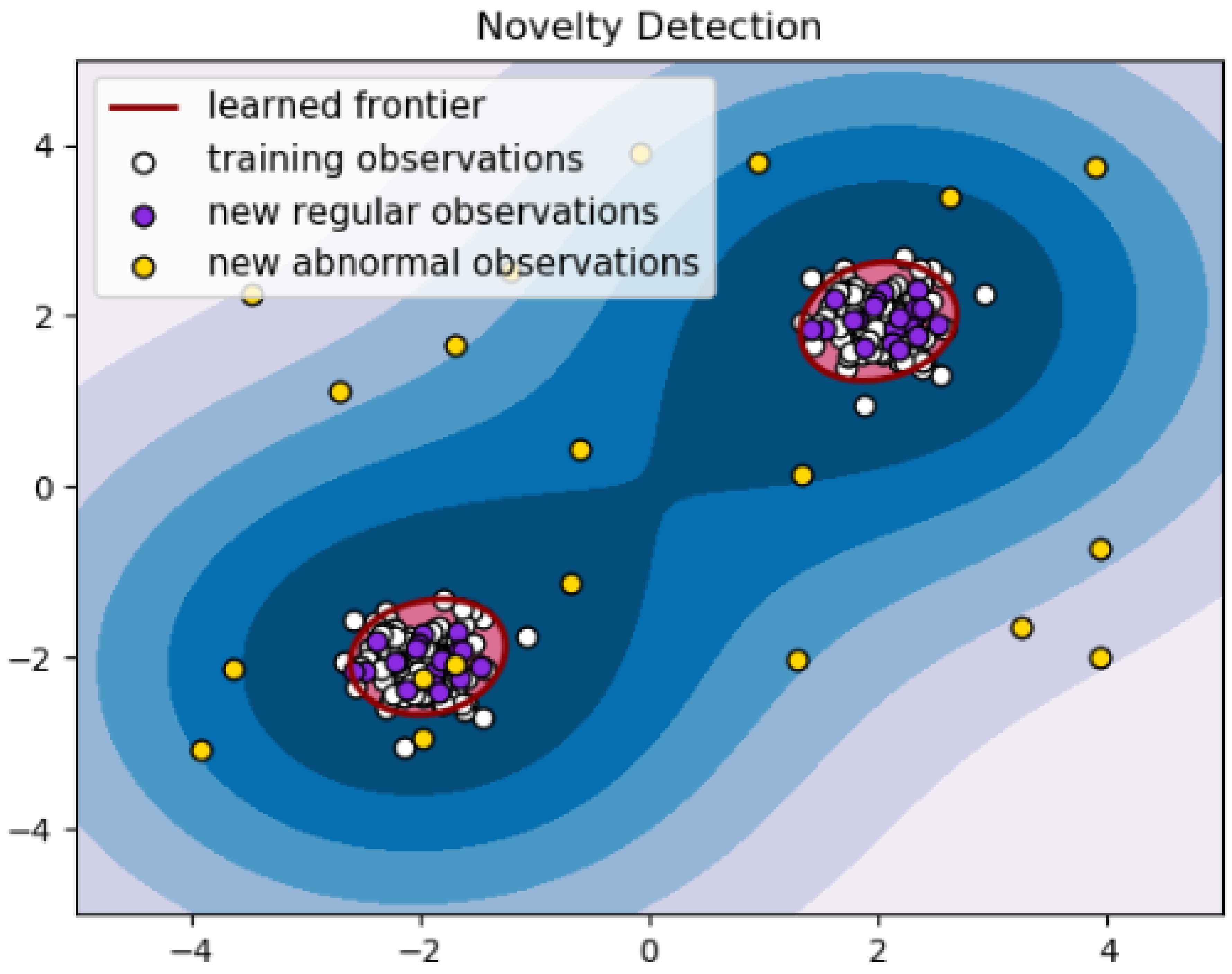
Practice 10

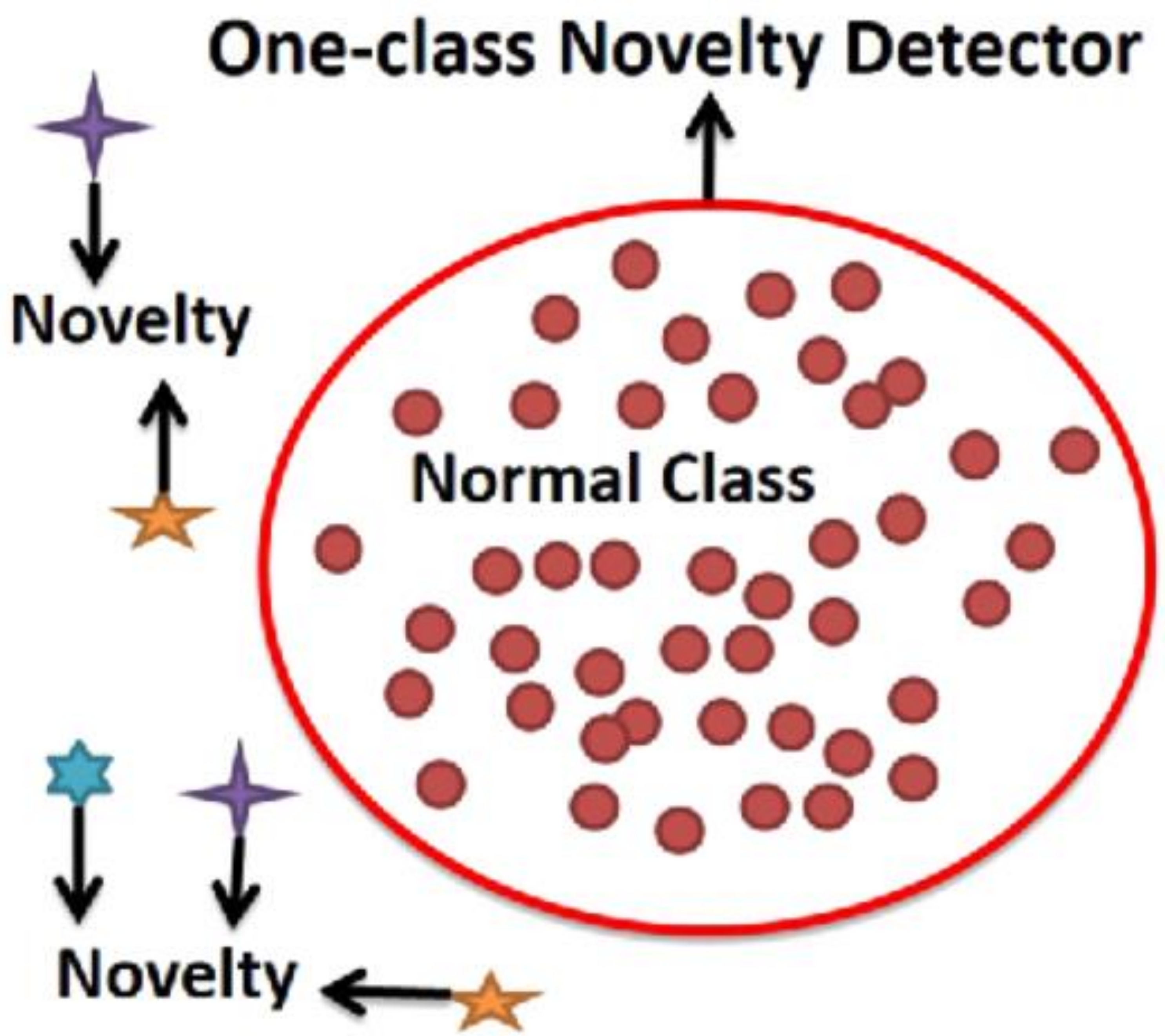
Novelty Detection

Introduction to Big Data Analytics(2020)

Problem

- Classify which data is normal or pathologic(outlier) using Novelty Detection with Spark.
- Use predefined ***sklearn.svm.OneClassSVM()***





https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html#sklearn.svm.OneClassSVM
https://www.semanticscholar.org/paper/LGND%3A-a-new-method-for-multi-class-novelty-Tang-Tian/2b18f73596e24b8587eed014f1c9f242e8e5f727/figure/0

Dataset

- **Cardiotocography**
  - The dataset consists of measurements of fetal heart rate and uterine contraction features on **cardiotocograms** classified by expert obstetricians
- **Explanation**
  - Data point has ***21 features, and label***
  - The label value was changed:

    From

    Label 0: ***inliers(normal)*** **data points**

    Label 1: ***outlier(pathologic)*** **data points**

    To

    Label 1: ***inliers(normal)*** **data points**

    Label -1: ***outlier(pathologic)*** **data points**

  - Because it is convenient to compare the predicted label of Novelty Detection and Real label

You can download dataset and see data description from below links.

http://odds.cs.stonybrook.edu/cardiotocogrpahy-dataset/
http://archive.ics.uci.edu/ml/datasets/Cardiotocography

Practice 10

1. **Calculate accuracy, and f1score of prediction using Novelty Detection algorithm to test data points and get Confusion matrix of the result.**

   ※ Note that training dataset is all normal, but test dataset is half normal and half abnormal.

   You can download this dataset on I-Campus.

   Please use dataset from I-Campus, not from UCI or Stonybrook.

2. **Use predefined classes in *sklearn.svm.OneClassSVM***

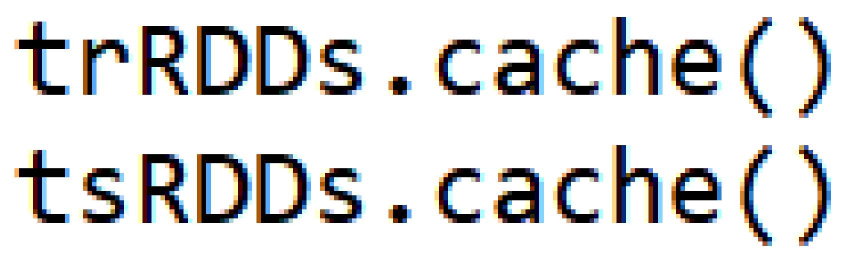
   Parameters for the method

   - **nu: 0.1, gamma: 0.1,kernel: 'rbf' (Don't change the other parameters)**

Practice 10

3. How to train the model using RDD data format

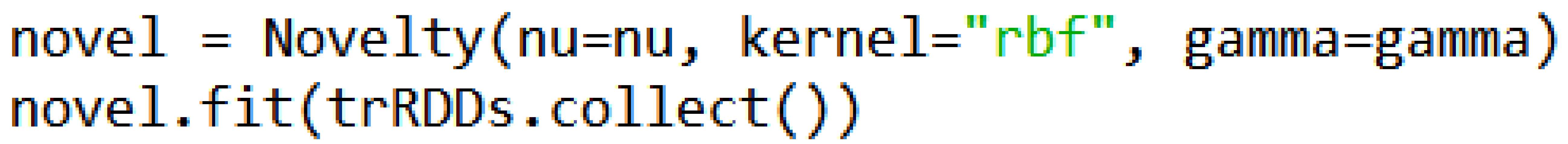
- Before training the model, you need to save data into your memory using ***cache()*** function.
- For example



```
trRDDs.cache()
tsRDDs.cache()
```

- In this example, ***trRDDs***: training data points & ***tsRDDs***: test data points

- Then, you can easily train NoveltyDetection model provided by scikit-learn using ***fit()*** function
- For example



```
novel = Novelty(nu=nu, kernel="rbf", gamma=gamma)
novel.fit(trRDDs.collect())
```

- In this example, ***nu & gamma***: parameters for Novelty Detection algorithm

https://spark.apache.org/docs/latest/rdd-programming-guide.html
https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

Practice 10

4. **After training the models, get the accuracy & F1 score for test data points**

5. **Get confusion matrix of the result**

6. **You need to use predefined arguments we suggest**

    - **Number of partitions: 30**

      You can split data when you make it RDDs.

      For example, “ ***RDD = sc.parallelize(Data, numPartition)*** ”

Submission

- ## You need to submit result.txt

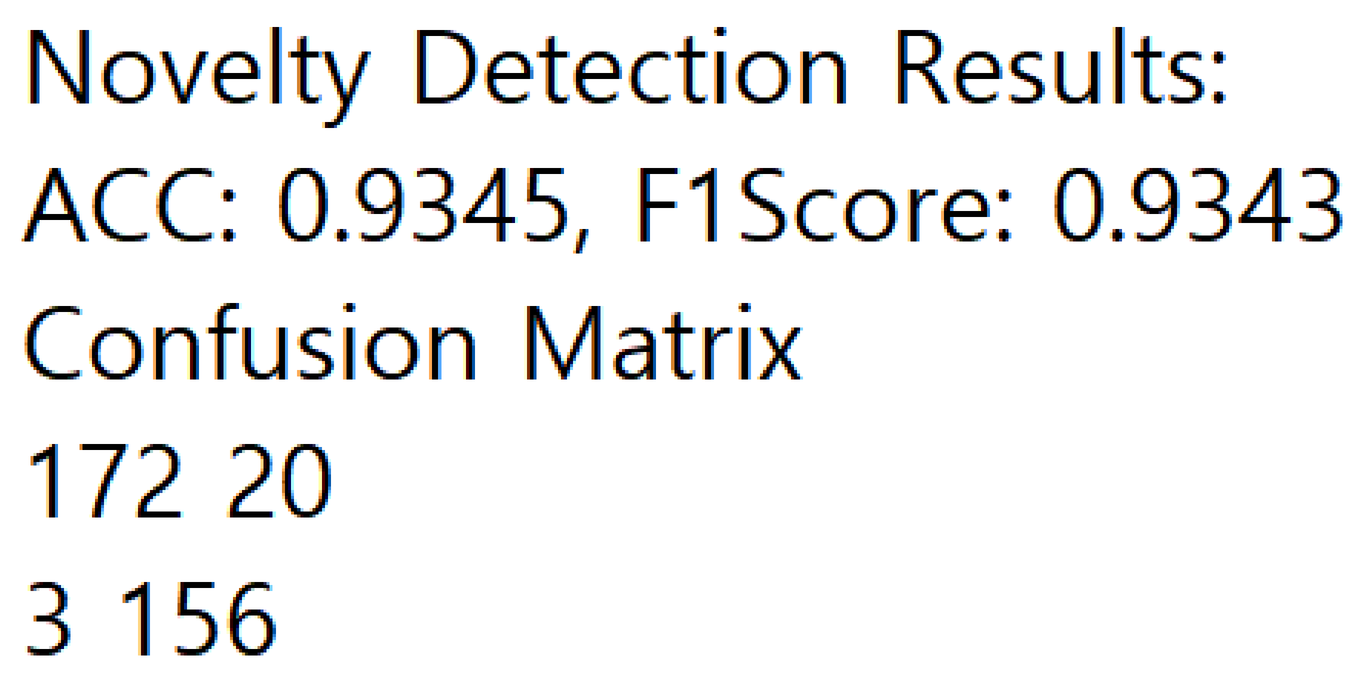
Write accuracy score of NoveltyDetection result, using ***sklearn.metrics.accuracy_score*** library

Then, write F1 score of NoveltyDetection result, using ***sklearn.metrics.f1_score*** library

Also, write confusion matrix of NoveltyDetection result, using ***sklearn.metrics.confusion_matrix*** library

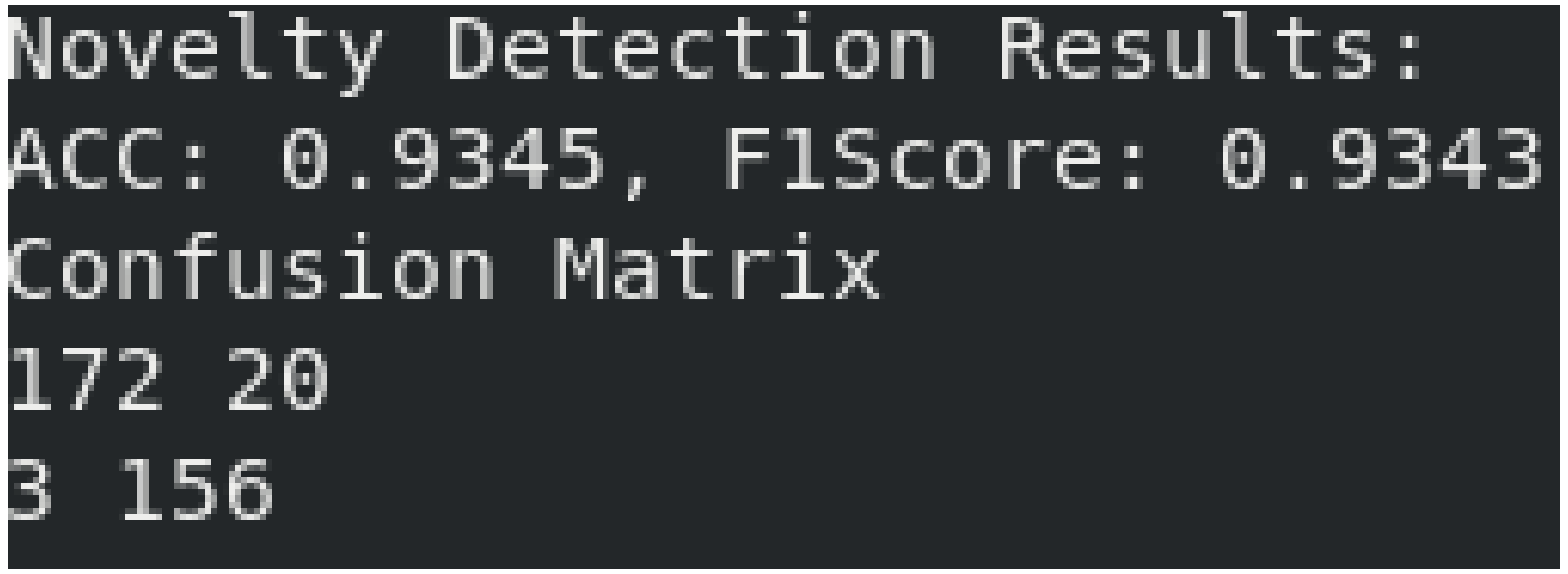
When you calculate F1 score, you need to use parameter **average = ‘macro’**

- ## Result



```
Novelty Detection Results:
ACC: 0.9345, F1Score: 0.9343
Confusion Matrix
172 20
3 156
```

Windows

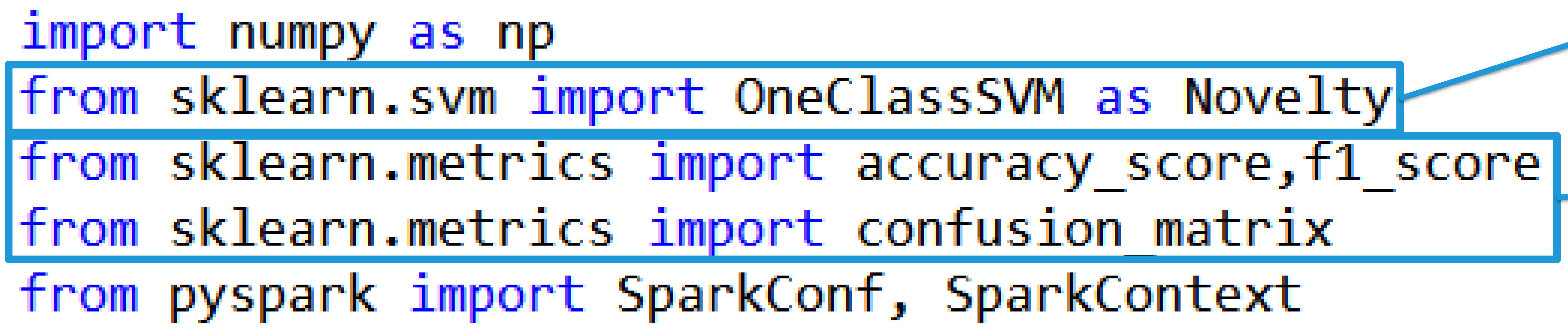


```
Novelty Detection Results:
ACC: 0.9345, F1Score: 0.9343
Confusion Matrix
172 20
3 156
```

Linux

Solution

- ➢ load libraries for Novelty Detection, and Spark Configuration

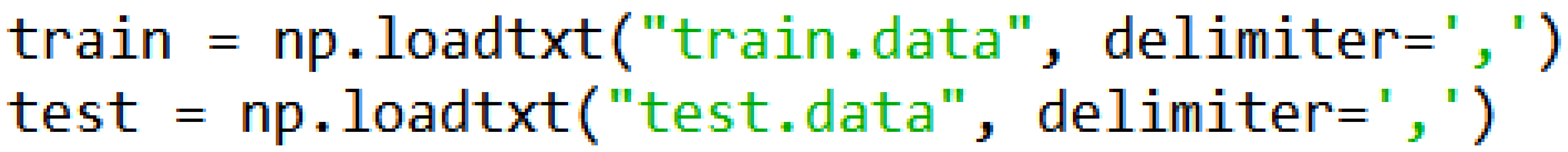


```
import numpy as np
from sklearn.svm import OneClassSVM as Novelty
from sklearn.metrics import accuracy_score,f1_score
from sklearn.metrics import confusion_matrix
from pyspark import SparkConf, SparkContext
```

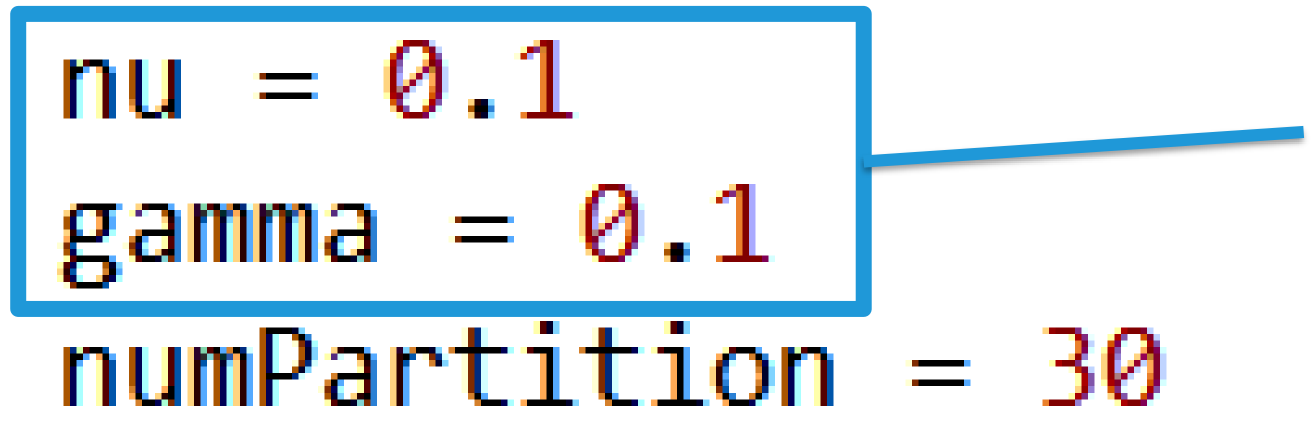
This library is for Novelty Detection

Using these library, you can calculate accuracy, f1score and get Confusion matrix

- ➢ Load dataset and set parameters



```
train = np.loadtxt("train.data", delimiter=',')
test = np.loadtxt("test.data", delimiter=',')
```

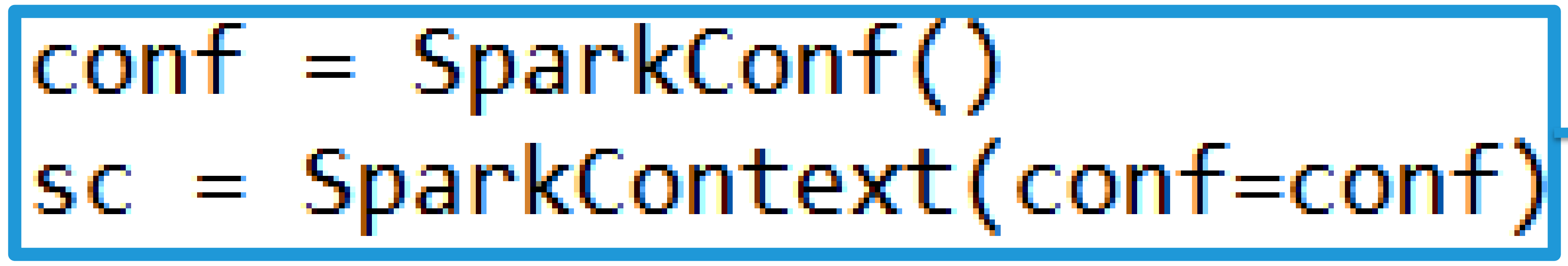


```
nu = 0.1
gamma = 0.1
numPartition = 30
```

To find out what these parameters mean, go to below link
https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

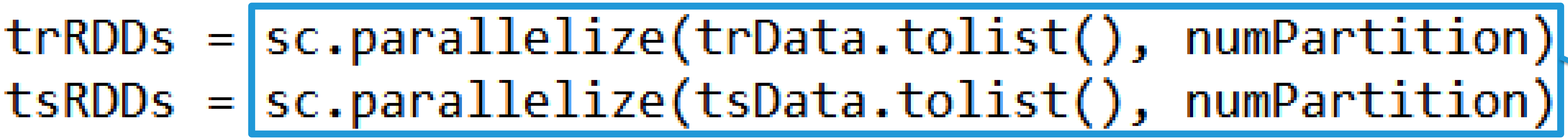
Solution

➢ Configure spark & make dataset to have RDD format



```
conf = SparkConf()
sc = SparkContext(conf=conf)
```

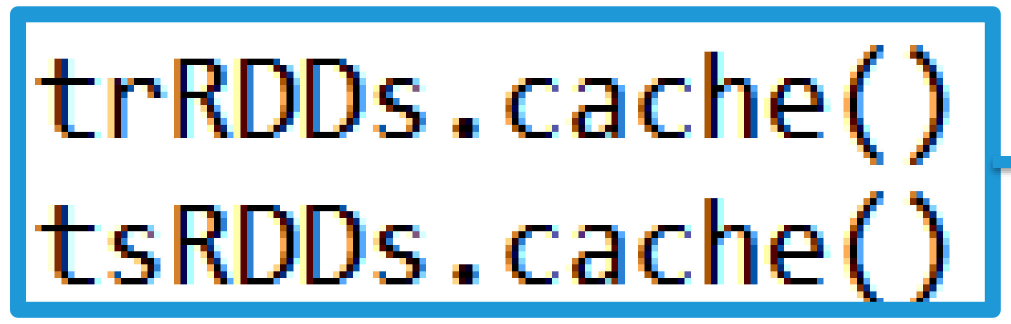
Configure Spark and define SparkContext



```
trRDDs = sc.parallelize(trData.tolist(), numPartition)
tsRDDs = sc.parallelize(tsData.tolist(), numPartition)
```

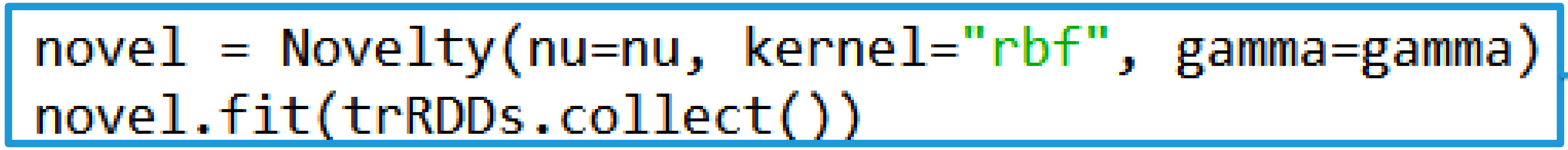
Make numpy.ndarray data to have RDDs format

➢ Save RDD to memory and Train the model



```
trRDDs.cache()
tsRDDs.cache()
```

Save data to memory

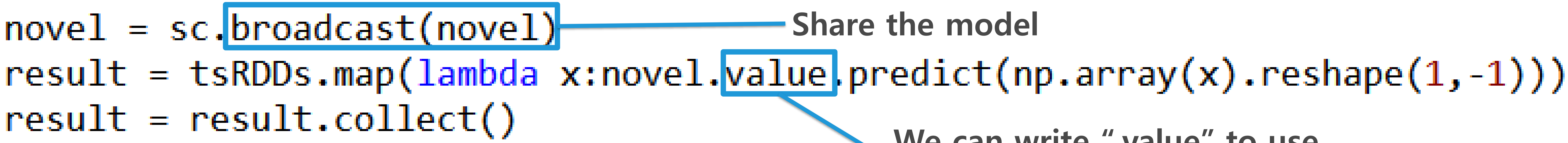


```
novel = Novelty(nu=nu, kernel="rbf", gamma=gamma)
novel.fit(trRDDs.collect())
```

Define model with given parameters and train model using fit() function

Solution

- ➢ Broadcast model & predict the label of test datapoints

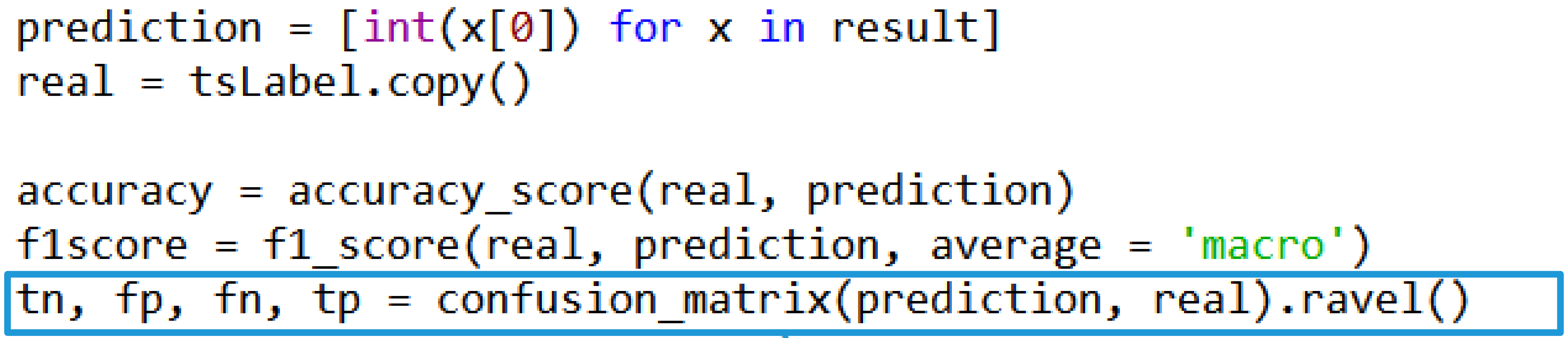


```
novel = sc.broadcast(novel)
result = tsRDDs.map(lambda x:novel.value.predict(np.array(x).reshape(1,-1)))
result = result.collect()
```

Share the model

We can write ".value" to use broadcasted model or variable

- ➢ Calculate accuracy and f1 score & Get confusion matrix of the result



```
prediction = [int(x[0]) for x in result]
real = tsLabel.copy()

accuracy = accuracy_score(real, prediction)
f1score = f1_score(real, prediction, average = 'macro')
tn, fp, fn, tp = confusion_matrix(prediction, real).ravel()
```

Using ".ravel()" we can get True Positive, False Positive, False Negative, True Positive value of Confusion Matrix

Solution

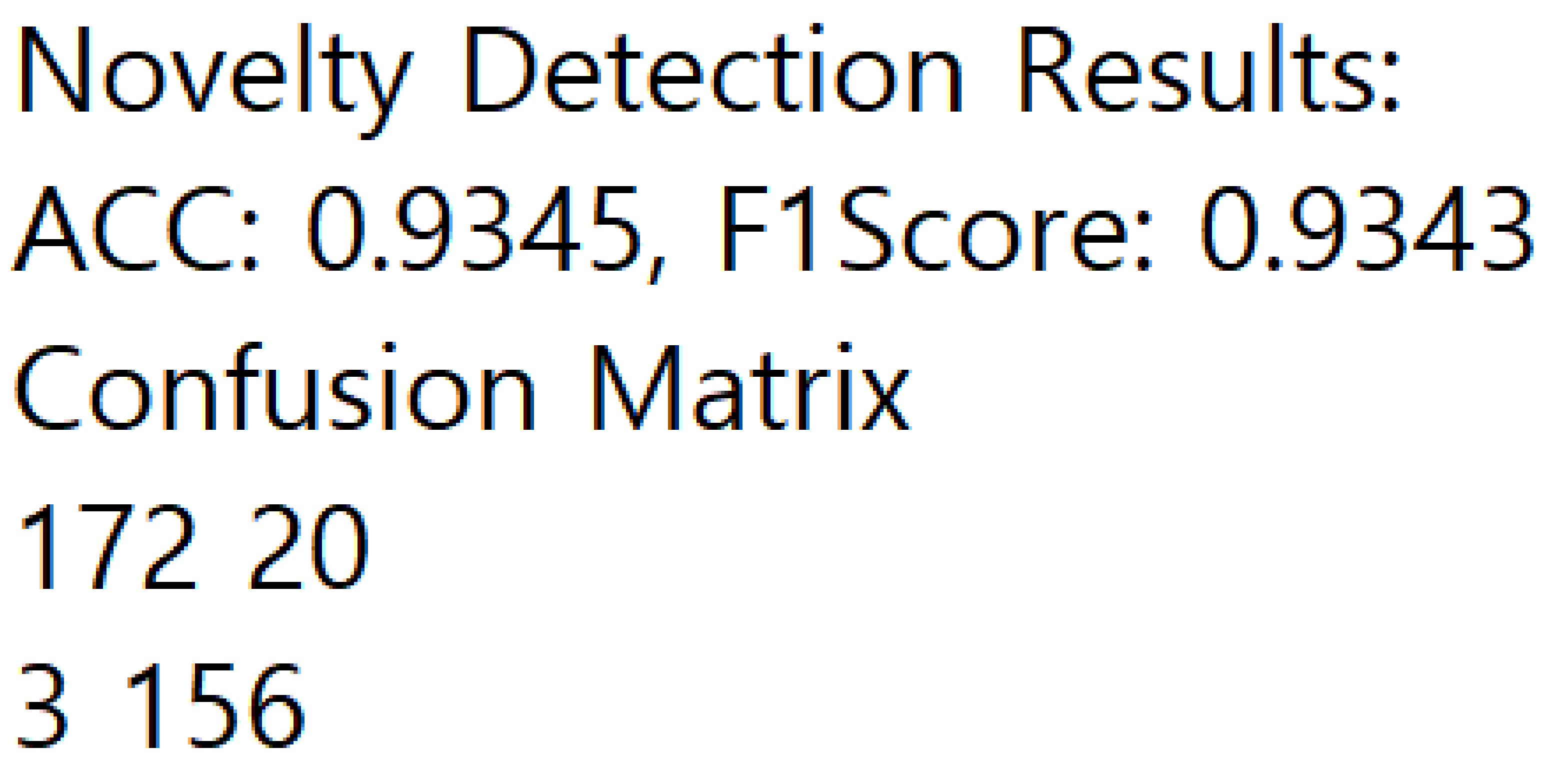
- Write the result to result.txt file & Stop Spark Context

```
f = open("result.txt", "w")
f.write("Novelty Detection Results:\n")
f.write("ACC: {:.4f}, F1Score: {:.4f}\n".format(accuracy,f1score))
f.write("Confusion Matrix\n")
f.write("{} {}\n".format(tn,fp))
f.write("{} {}\n".format(fn,tp))

sc.stop()
```

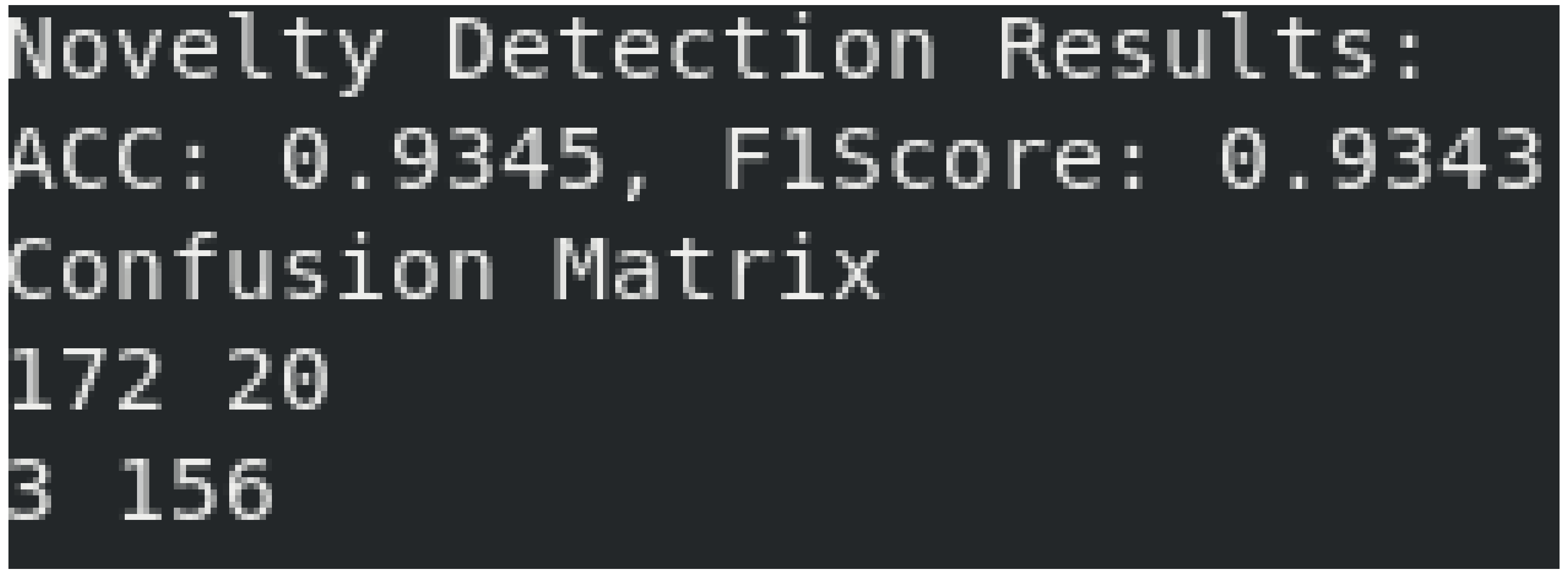
You must stop spark context after finishing your code. If you don't another spark can get wrong result

- Result



```
Novelty Detection Results:
ACC: 0.9345, F1Score: 0.9343
Confusion Matrix
172 20
3 156
```

Windows



```
Novelty Detection Results:
ACC: 0.9345, F1Score: 0.9343
Confusion Matrix
172 20
3 156
```

Linux