# One-class Classification

## BITS F464 Machine Learning

## *Assignment 1*

***Submitted By:***

Aditya Agrawal : 2020B5A42010P

Niranjan Chaudhari: 2020B5A30929P

Rahul James: 2020A2PS1334P

***Submitted to*:**

Prof. Navneet Goyal,

CSIS Department

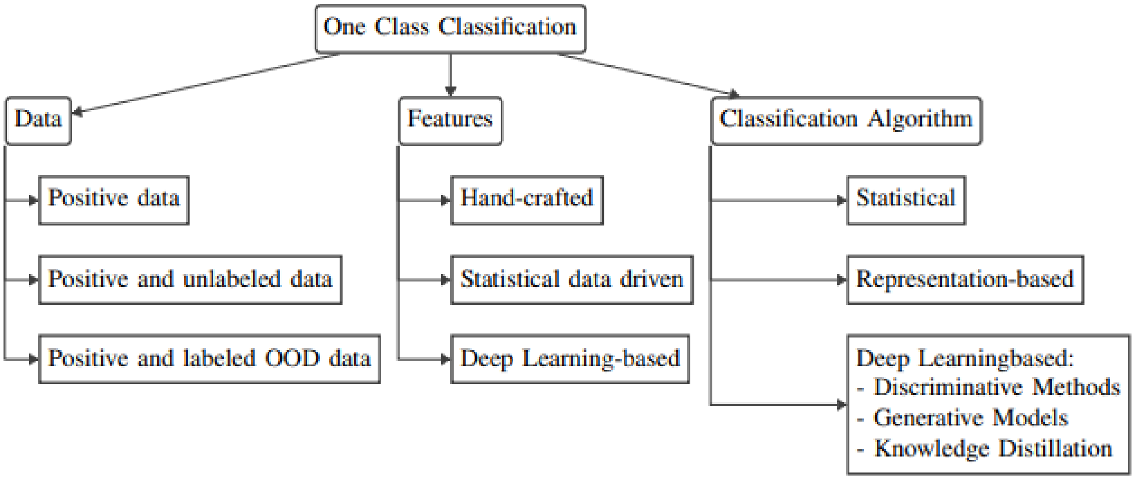
BITS Pilani

# One-class classification

One-class classification is a particular area of machine learning that focuses on locating outliers or unusual occurrences within a dataset. It is often referred to as an anomaly or outlier identification. One-class classification algorithms seek to model the "normal" examples found in the data and categorize new cases as either normal or abnormal, in contrast to standard classification techniques that require data from several classes.

Because these algorithms can be trained on instances from the majority class and subsequently assessed on a holdout test dataset, they benefit binary classification tasks with significantly skewed class distributions. They may also be helpful in unbalanced datasets with few samples from the minority class or that lack a distinct structure that would allow classes to be distinguished.

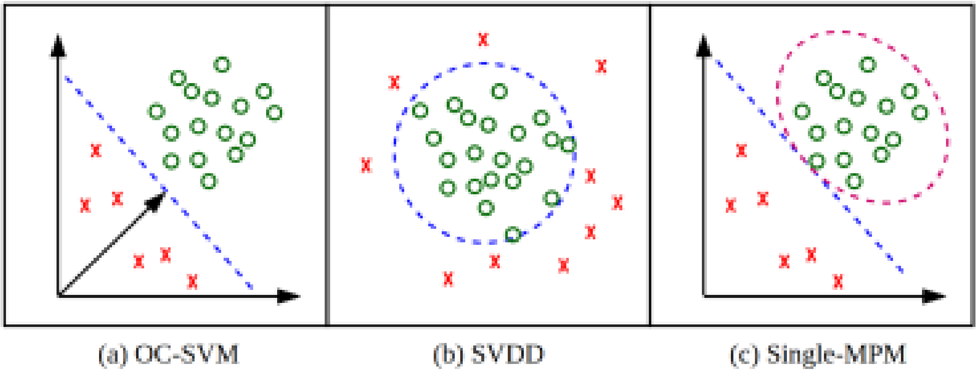
Numerous techniques, such as those based on Support Vector Machines (SVM), Deep Convolutional Neural Networks (DCNNs), and other statistical methods, have been presented for one-class classification. SVM-based methods like Support Vector Data Description (SVDD) and One-Class SVM (OC-SVM) aim to enclose the target class data in a hypersphere or maximize the margin between the target class and the origin.



[4] discusses the taxonomy of OCC.

Due to DCNNs' remarkable performance in object identification and recognition tasks, there has been an increasing amount of interest in employing them for one-class classification in recent years. However, because there isn't any negative class data, training DCNNs for one-class issues can be difficult. To solve this problem, generative techniques have been proposed, such as auto-encoders and generative adversarial networks (GANs), which learn unique features for the target class.

One-class classification discriminative techniques that use new loss functions or outside reference datasets have also been investigated. With the use of these techniques, DCNNs should be trained end-to-end to acquire representations that are especially suited for one-class situations.


(a) OC-SVM

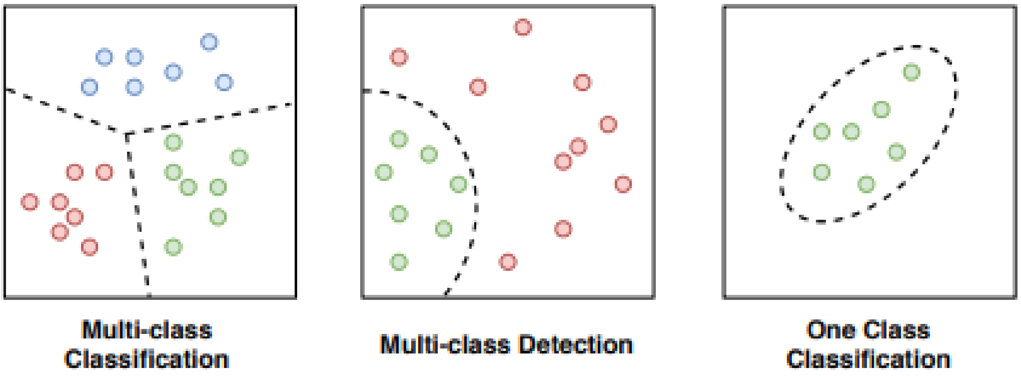
(b) SVDD

(c) Single-MPM

Popular statistical one-class classification methods (Obtained from [5])

All things considered, one-class classification methods are pretty useful for anomaly, event, and biometric detection, and they are also essential for locating outliers or abnormalities within datasets. Promising progress has been made recently in deep learning-based methods to tackle the difficulties involved in one-class classification tasks.

The report discusses isolation forests, one class SVM, and local outlier factors. We have implemented three algorithms for one class classification and provided the parameters for comparison. Followed by the advantages and disadvantages of various algorithms in OCC.



Modes of classification  (Obtained from [4])

## How are they different from binary classification problems?

Binary classifiers have long been the industry standard for building classification models in the field of machine learning. On the other hand, one-class classification (OCC) deviates from this standard by concentrating only on creating models using a single class of data. This is especially helpful in situations where a single class has a large amount of data. When there is a large disparity in the number of students in two classes, standard binary classifiers may be unable to classify students accurately. One-class classifiers provide a workable answer in these situations.

In one of the research papers [2], the performance of one-class and binary classifiers with increasing imbalance results in increased uncertainty in the second class. The goal is to shed light on which classification paradigm is more appropriate as imbalances and uncertainty increase. To accomplish this, experiments were run on a variety of datasets, derived from the UCI repository as well as synthetic data. The steady decrease in the size of the second class, increased the degree of imbalance, while closely observing the performance of binary and one-class classifiers. The findings demonstrate that while one-class classifiers remain comparatively stable, binary classifiers perform worse as the degree of imbalance rises.

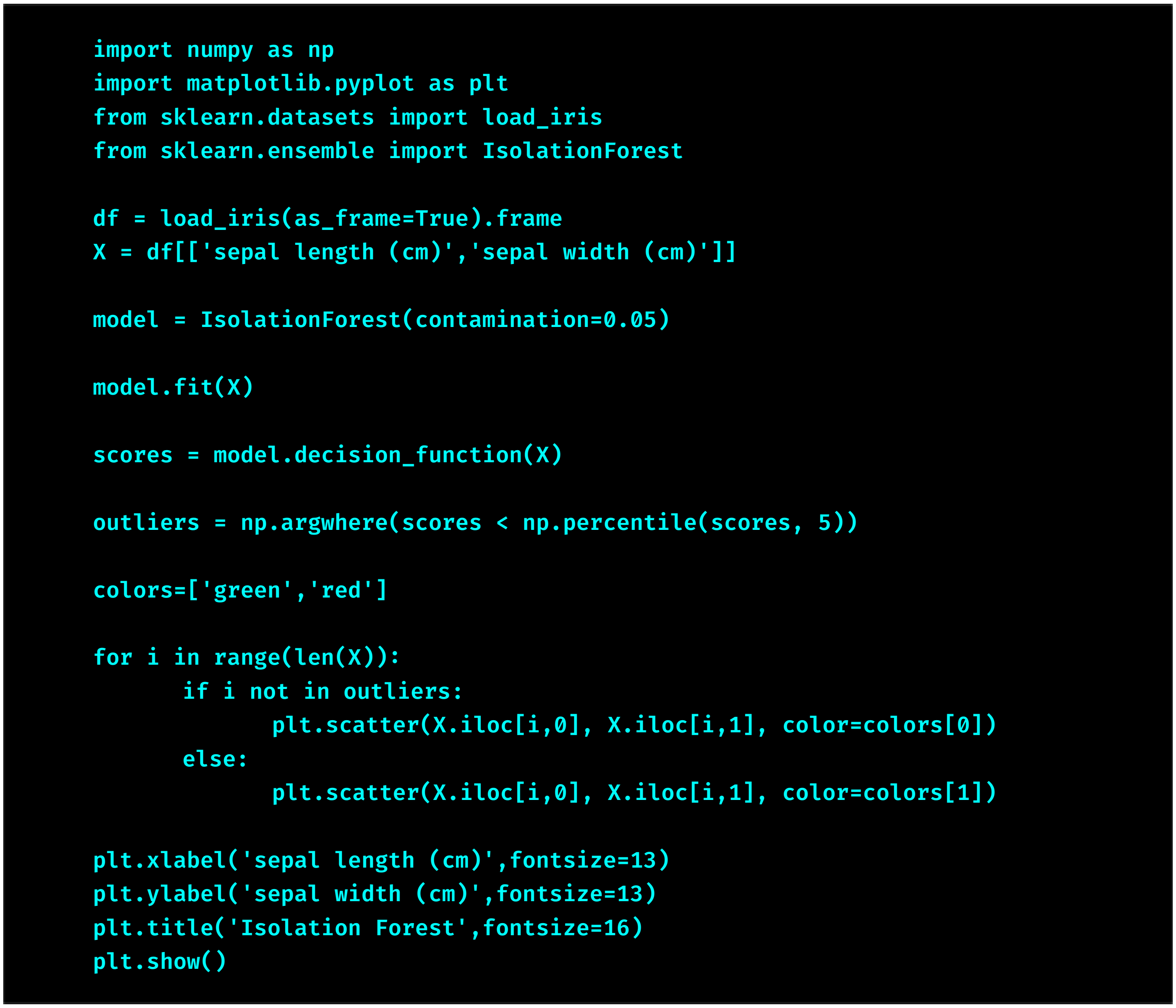
Given the significant financial consequences connected to fraudulent activity, machine learning research must prioritize fraud detection. One of the research papers ([3]) reviewed focuses its analysis on two datasets that are well known for having a significant class imbalance: the Medicare Part D dataset and the Credit Card Fraud dataset. The Credit Card Fraud Detection Dataset is a valuable resource for assessing credit card fraud detection methods due to its large amount of transactional data. Conversely, the Medicare Part D dataset, with its wide range of coverage, provides information about national trends and patterns regarding prescription drug use and costs. The above evaluated the performance of several classifiers, such as ensembles of decision trees, logistic regression, One-Class SVM, One-Class GMM, and OCAN, by using different techniques. According to findings, binary classification outperforms one-class classification in detecting fraud incidents in severely unbalanced data.

CatBoost had the greatest results for binary classification, with AUPRC scores of 0.8567 and 0.8124 for the Medicare Part D dataset and the credit card fraud detection dataset, respectively. OCC learners may have a more difficult time identifying the minority class given the significant difference in AUPRC outcomes between the binary and OCC algorithms. Binary classification is advised if both class labels are easily obtained. On the other hand, OCC is advised to use only one class label if it is readily available.

## Three algorithms for solving One-class Classification algorithms and applying them to the applications identified in 1.

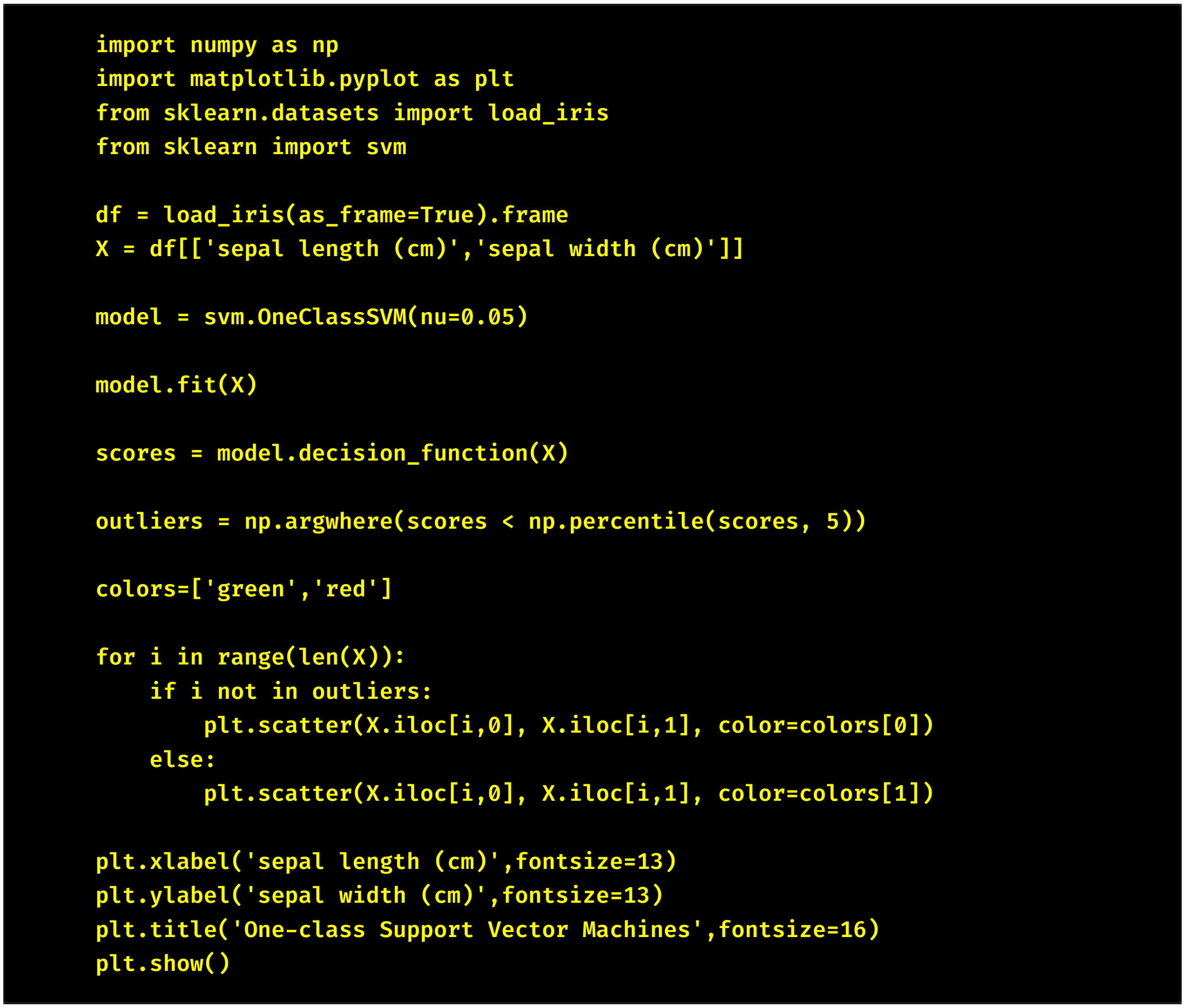
We obtained a basis for comparing three widely used algorithms for One-class Classification problems using the following codes. [7]

**Isolation Forest** :



```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.ensemble import IsolationForest

df = load_iris(as_frame=True).frame
X = df[['sepal length (cm)','sepal width (cm)']]

model = IsolationForest(contamination=0.05)

model.fit(X)

scores = model.decision_function(X)

outliers = np.argwhere(scores < np.percentile(scores, 5))

colors=['green','red']

for i in range(len(X)):
       if i not in outliers:
              plt.scatter(X.iloc[i,0], X.iloc[i,1], color=colors[0])
       else:
              plt.scatter(X.iloc[i,0], X.iloc[i,1], color=colors[1])

plt.xlabel('sepal length (cm)',fontsize=13)
plt.ylabel('sepal width (cm)',fontsize=13)
plt.title('Isolation Forest',fontsize=16)
plt.show()
```

## One class SVM :



```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn import svm

df = load_iris(as_frame=True).frame
X = df[['sepal length (cm)','sepal width (cm)']]

model = svm.OneClassSVM(nu=0.05)

model.fit(X)

scores = model.decision_function(X)

outliers = np.argwhere(scores < np.percentile(scores, 5))

colors=['green','red']

for i in range(len(X)):
    if i not in outliers:
        plt.scatter(X.iloc[i,0], X.iloc[i,1], color=colors[0])
    else:
        plt.scatter(X.iloc[i,0], X.iloc[i,1], color=colors[1])

plt.xlabel('sepal length (cm)',fontsize=13)
plt.ylabel('sepal width (cm)',fontsize=13)
plt.title('One-class Support Vector Machines',fontsize=16)
plt.show()
```

## Local Outlier Factor (LOF) :

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.neighbors import LocalOutlierFactor

df = load_iris(as_frame=True).frame
X = df[['sepal length (cm)','sepal width (cm)']]

lof = LocalOutlierFactor(n_neighbors=5)

lof.fit(X)

scores = lof.negative_outlier_factor_

outliers = np.argwhere(scores > np.percentile(scores, 95))

colors=['green','red']

for i in range(len(X)):
    if i not in outliers:
        plt.scatter(X.iloc[i,0], X.iloc[i,1], color=colors[0])
    else:
        plt.scatter(X.iloc[i,0], X.iloc[i,1], color=colors[1])

plt.xlabel('sepal length (cm)',fontsize=13)
plt.ylabel('sepal width (cm)',fontsize=13)
plt.title('Local Outlier Factor',fontsize=16)
plt.show()
```

## Comparison of algorithms

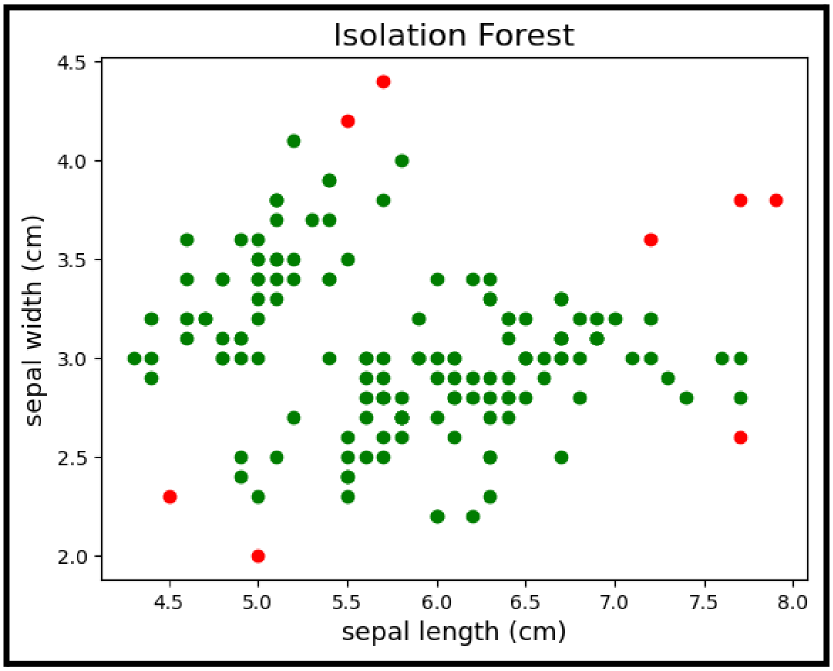
A **quantitative** comparison of the 3 algorithms used can be made using the accuracy score. In the above cases, the outliers were decided on the basis of 95th percentile. Outliers are those data points that do not fall in the 95th percentile of the score. The models are run over the IRIS dataset.

| Algorithm | Anomalies Present | Anomalies Detected |
|---|---:|---:|
| Isolation Forests | 8 | 8 |
| One Class SVM | 8 | 8 |
| Local Outlier Factor | 7 | 4 |

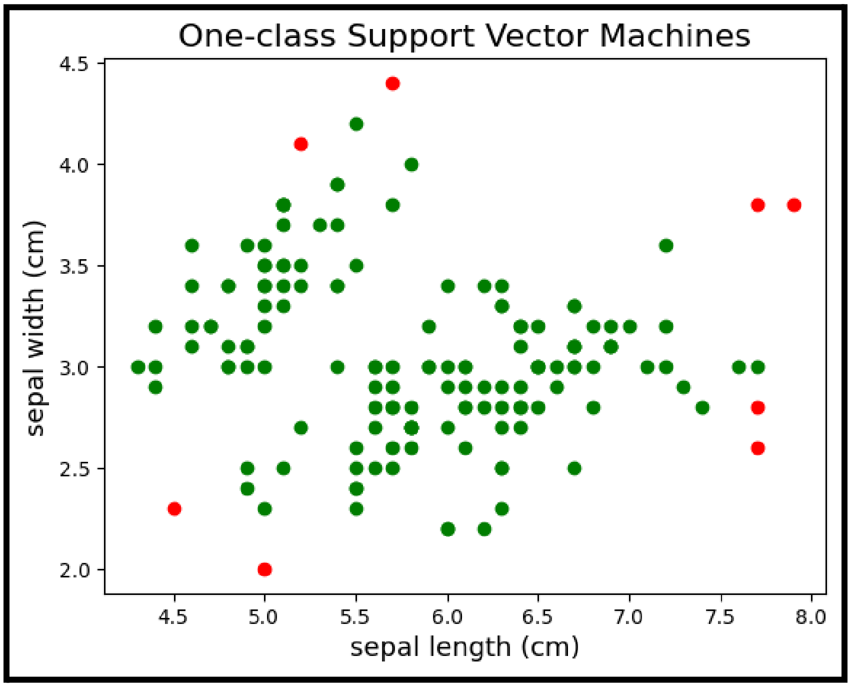
On the **qualitative front**, there can be some data points that can be misclassified (False positives and false negatives).

The graphical representation of the above algorithms is as follows:
(The red dots show the misclassified points while the green ones are correctly classified)

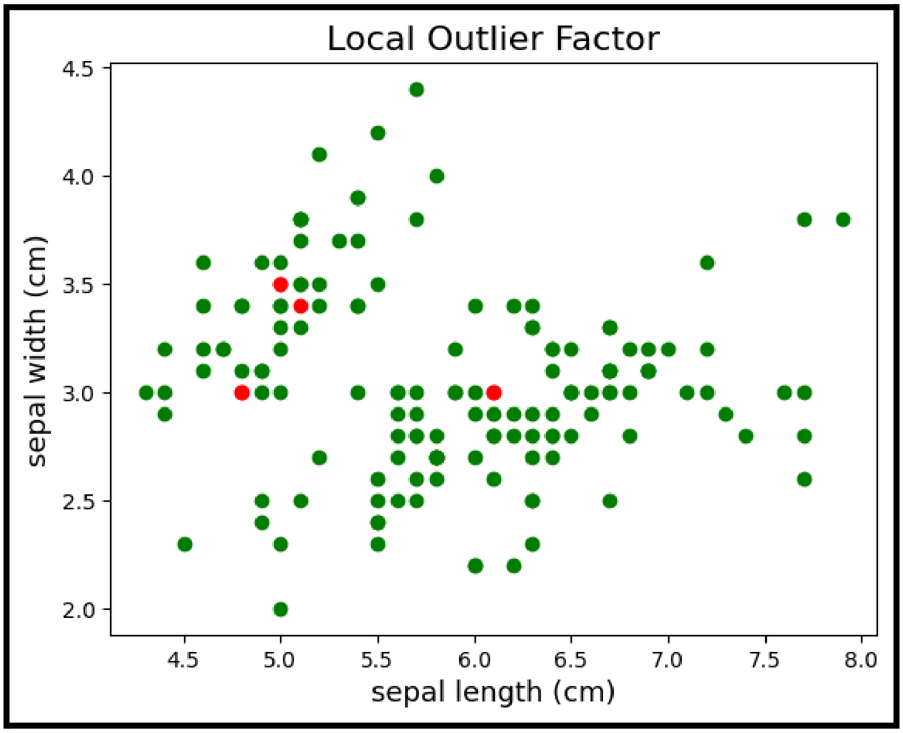
**Isolation Forests** :



**One-class SVM** :



**Local Outlier Factor** :



Now, discussing a few advantages and disadvantages,

**Isolation Forests** :

Advantages:

- Isolation forests are effective in detecting anomalies and outliers in high-dimensional datasets.
- They are less sensitive to the choice of parameters and can handle both categorical and numerical features.

Disadvantages:

- Isolation Forests are not optimized for multi-class classification tasks and treat all anomalies as outliers, which may not be appropriate for certain scenarios.

**One-class SVM** :

Advantages:

- One-class SVM is effective in handling high-dimensional data and capturing complex decision boundaries.
- It is suitable for anomaly detection and binary classification tasks, providing a robust solution in these scenarios.

Disadvantages:

- One-Class SVM requires careful selection of kernel functions and associated parameters, which can be challenging and impact performance.
- It is primarily designed for binary classification or anomaly detection and may not be easily adapted to multi-class classification problems.

**Local Outlier Factor** :

Advantages:

- One-class SVM is effective for anomaly detection in datasets with high-dimensional and complex structures.
- It does not make assumptions about the data distribution and can capture non-linear relationships between data points.

Disadvantages:

- One-class SVM can be computationally expensive for large datasets and is sensitive to the choice of hyperparameters, such as the kernel and the regularization parameter.
- It may also struggle with imbalanced datasets or when the proportion of outliers is very low.

# References

/1) One-Class versus Binary Classification: Which and When? | IEEE Conference Publication

2) https://www.researchgate.net/publication/261156791_One-Class_versus_Binary_Classification

3) Investigating the effectiveness of one-class and binary classification for fraud detection | Journal of Big Data

4) One-Class Classification: A Survey

5) One-Class Convolutional Neural Network

6) One-Class Classification Algorithms for Imbalanced Datasets - MachineLearningMastery.com

7) Comparing anomaly detection algorithms for outlier detection on toy datasets in Scikit Learn