1．实验目的

（1）理解Isolation Forest算法原理

（2）理解Local Outlier Factor算法原理

（3）掌握离群点检测的无监督方法。

（4）能够正确评价算法结果

2. 算法说明

（1）LOF算法（Local Outlier Factor），又叫局部异常因子，是一种基于距离的、无监督的离群点检测算法。

class sklearn.neighbors.**LocalOutlierFactor**(n\_neighbors=20, \*, algorithm='auto', leaf\_size=30, metric='minkowski', p=2, metric\_params=None, contamination='auto', novelty=False, n\_jobs=None)

参数contamination，相当于离群点在数据集中的占比，用作阈值

* if ‘auto’, the threshold is determined as in the original paper,
* if a float, the contamination should be in the range [0, 0.5].

代码示例：

# fit the model for outlier detection (default)

clf = LocalOutlierFactor(n\_neighbors=20, contamination=0.1)

y\_pred = clf.fit\_predict(X) #LOF没有predict方法

X\_scores = clf.negative\_outlier\_factor\_

（2）孤立森林算法（Isolation Forest ），是一种无监督的离群点检测算法，可发现全局离群点。

class sklearn.ensemble.**IsolationForest**(\*, n\_estimators=100, max\_samples='auto', contamination='auto', max\_features=1.0, bootstrap=False, n\_jobs=None, behaviour='deprecated', random\_state=None, verbose=0, warm\_start=False)

代码示例

# fit the model

clf = IsolationForest(max\_samples=100, random\_state=rng)

clf.fit(X)

y\_pred = clf.predict(X)

3．实验要求和步骤

（1）读取KDD Cup网络入侵数据kddcup.data\_10\_percent，将非数值字段转码，并完成数据预处理。要求将类别标签中各种攻击类型，修改为“异常”类别，数据集转换为“正常”和“异常”两个类别。

（2）加载数据到变量X中，X=预处理并去除类别标签后的数据。类别标签仅用于模型评估。

（3）分别使用Isolation Forest算法和Local Outlier Factor算法，训练模型，并输出预测结果y\_pred，根据已知的类标标签，评估模型的性能。

计算错误率的Python示例代码为：

n\_errors = (y\_pred != true\_label).sum()

（4）优化模型参数，观察预测准确率是否提高。

按步骤截图，添加必要的说明文字

特别是采取某个处理策略时，应说明“理由和处理的合理性”

4．分析与讨论

（1）讨论还有哪些算法，适合分析预测网络入侵数据。

回答问题，记录个人知识总结、调试经验等

5．附录

附实现代码