Other techniques for finding anomaly/novelty/outlier detections are covered in this lecture.

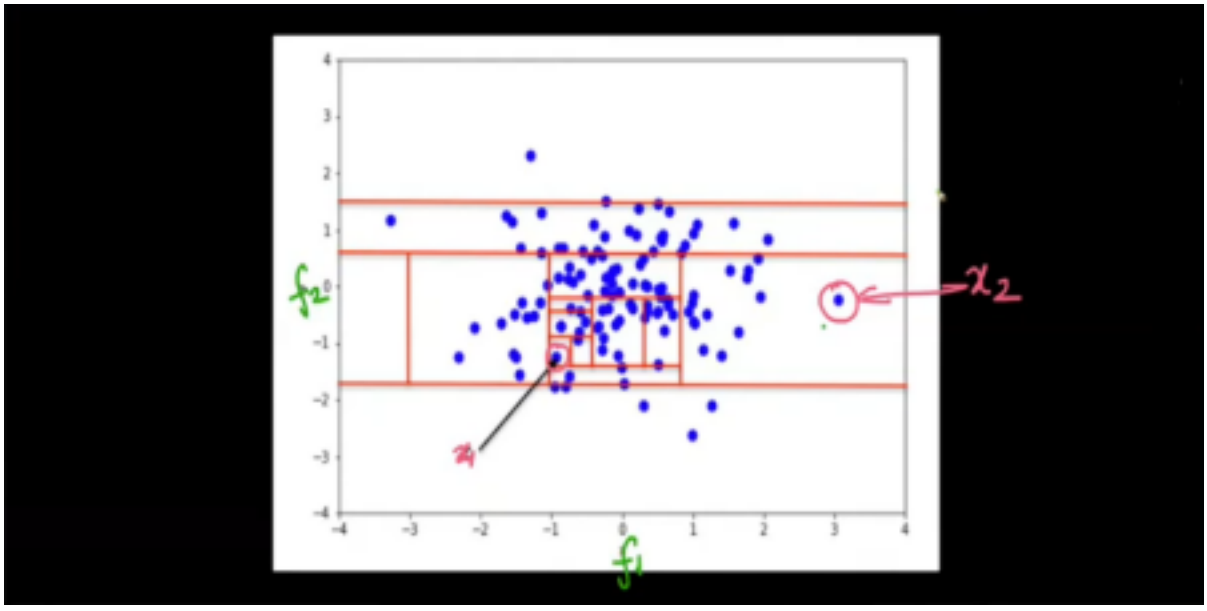
1. Isolation Forests

- Consider a dataset D which contains data points x_1, x_2,...., x_n. Just like Random forests, Isolation Forests build many trees.

- Following are the steps involved in Isolation Forest:
    - Build many trees like random forests
    - For each tree:
        - Randomly pick a feature
        - Randomly threshold that features
        - Build each tree until the leaf consists of only one datapoint

- Isolation Forests are also known as iForests

- Consider the plot along feature f_1 and f_2 given below:

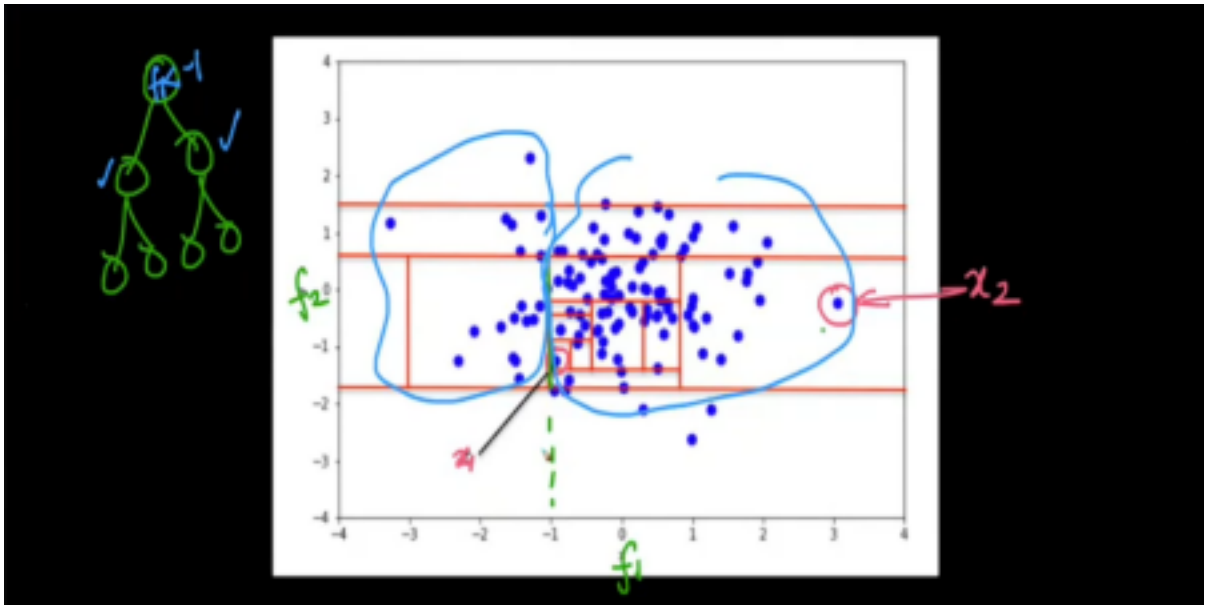


- In isolation forests, we are building totally random trees. So if we pick feature f_1 and put a threshold there will be a vertical bar.

- Similarly, if we pick feature f_2 and put a threshold there will be a horizontal bar.

- For example, if we pick feature f_1 and we select threshold as $f_1 < 1$, then our first

root node will be based on this condition



- Based on the diagram above,
  - The node containing x_1 will be at more depth.
  - Observe that the point x_1 is in a dense region, and point x_2 is far away
  - That is because, to break the point x_1 from all the other points, more and more splits will be required and that will increase the depth of the node containing point x_1.
- So, to sum it up, the idea behind Isolation forest is:
  - On average outliers have lower depth in the random trees
  - On average, inliers have more depth in the random trees

Evaluation of Isolation Forest

- Imagine, we have to build random trees. For each point x_i in the dataset, we can get an average depth.
- We use this average depth to convert it into a metric.
- Apart from this, there are a lot of different metrics, that people have come up with over the years
- But, the basic intuition is that the lesser the average depth, the higher the likelihood is there that it is an outlier

Deciding average depth of a point:

- There are a lot of metrics that researchers have come up with over the years.

- But, studying them in this lecture is out of scope.

Sklearn walkthrough

We can implement Isolation Forest with the help of sklearn's **IsolationForest** method present in the **ensemble** module.

Let's see some of the parameters that IsolationForest expects:

1. **n_estimators**: It represents the number of base learners. By default, the value is set equal to 100

2. **max_sample**: It is the number of samples to extract from the dataset to build the trees(row sampling). By default the value is set to auto and sklearn picks reasonably a good figure for iForests

3. **contamination**: It tells the proportion of outliers in the data. The range is between [0,0.5]

4. **max_features**: It is the number of features to extract from the dataset to build the trees(column sampling).

Disadvantages

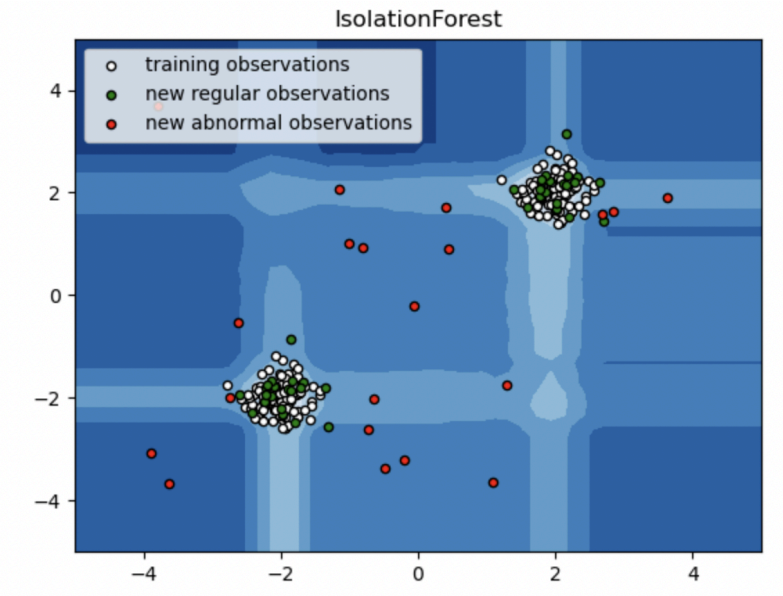
- One of the major limitations of iForests is that they are biased towards axis parallel splits.

- iForests makes splits and these splits are always parallel to either of the axis.

- Because of this, the boundary will not be smoothened.
    - In the diagram given below, the different shades of blue represent the likelihood of a point is an outlier. The darker the color, it is more likely that the point in that region will be an outlier

    - We've trained the iForest model using training data(white points)

    - It is tested on testing data(red + green) where red color indicates outliers

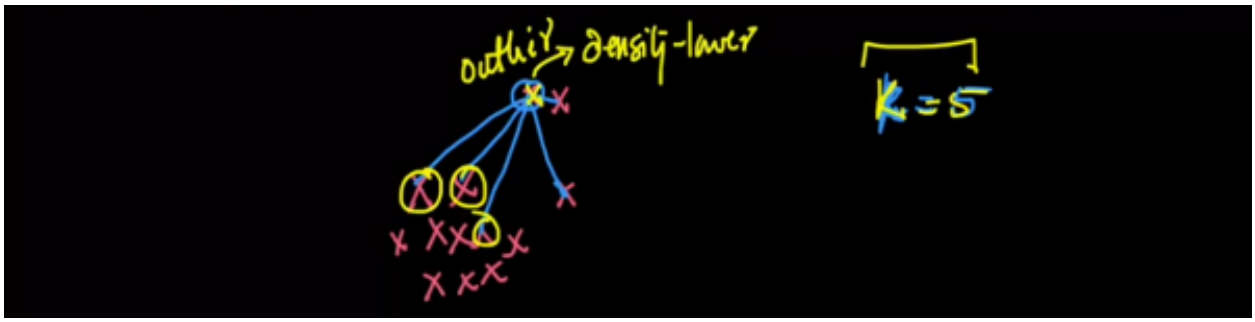


- Now imagine two points x_1 and x_2 as shown in the diagram given below.
- Both the points are almost equidistant from the nearest cluster. x_1 is on the axis and point x_2 is off-axis.
- Because the model is biased towards the axis, it will classify the point as an inlier and as an outlier
- This is also known as banding in signal processing

2. Local Outlier Factor (LOF)

- On a higher level, LOF is based on two ideas: KNN and density
- The core idea behind LOF is to compare the density of a point with its neighbors' density
- If the density of a point is less than the density of its neighbors, we flag that point as an outlier

- Imagine a bunch of datapoints as shown below



- We compute the density of a point based on average distance.
- If the average distance between a point and it's **K** nearest neighbors is large, it is more likely that the point will be an outlier
- Also, the larger the value of **K**, the more confident are the results.

Some concepts to understand the working of LOFs:

1(a) K-distance

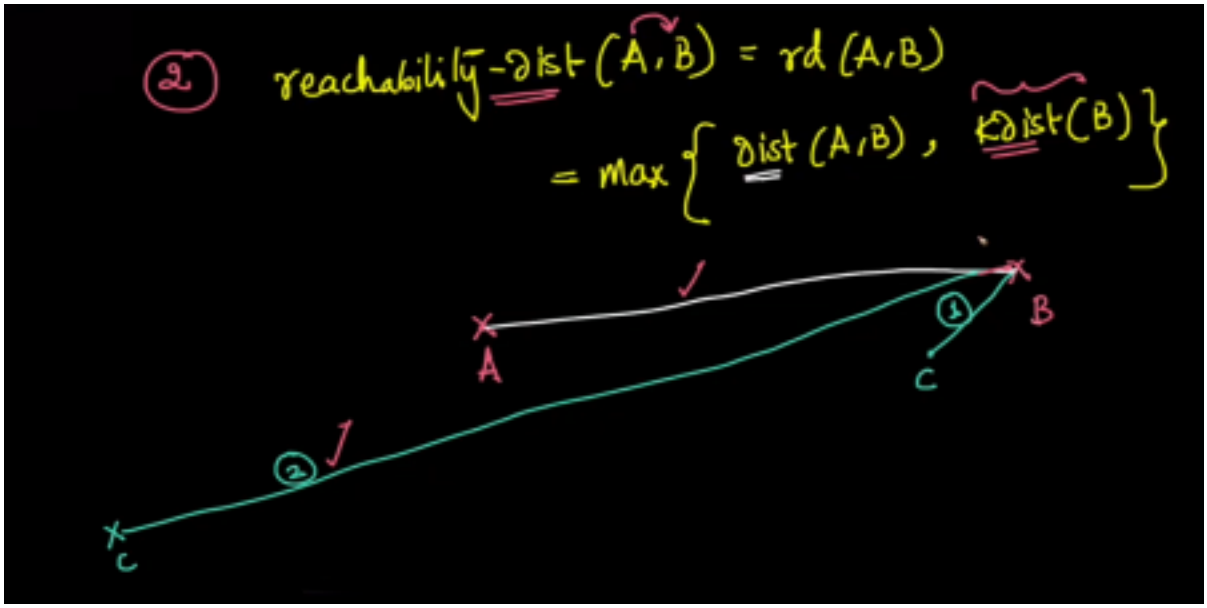
- We define K-distance of a point **A** as the distance of point **A** to its **K^{th}** nearest neighbor
- In general, the larger the value of k-distance is, the farther away the point is from other datapoints

1(b) Set: Nk (A)

- It is a set of k-nearest neighbors of point **A**.

2. Reachability distance

- From point **A** to point **B**, we define reachability distance as a maximum of the distance from point **A** to point **B** and the maximum k-distance of point **B**
- Consider point **B** with some **k** nearest neighbors shown in the diagram below.



- There is a possibility that some neighbors might be close(condition 1) and some neighbors might be very far away(condition 2)
- In this case, there is a neighbor of point **B** whose k-distance is greater than the distance between point **A** and **B**, and hence, it is considered as its reachability distance.

3. Local Reachability Density

- It is often represented as **$lrd_k(A)$**, which tells the local reachability density of a point **A**
- It is defined as the average reachability distance between point **A** and **k** neighbors

$$\text{So, } lrd_k(A) = \frac{\sum_{B \in N_k(A)} rd_k(A,B)\$}{N_k(A)}$$

-
- The summation in the above equation represents the sum of reachability distances from a point **A** and set of neighbors **B** as **B** $\in$ **$N_k(A)$**

- We define Local Outlier Factor of point as follows:

$$LOF_k(A) = \frac{\sum_{B \in N_k(A)} lrd_k(B)}{|N_k(A)|.ldr_k(A)}$$

- **$lrd_k(A)$** is the density of point **A**
- The expression $\frac{\sum_{B \in N_k(A)} lrd_k(B)}{|N_k(A)|}$ is the average neighborhood density
- So, LOF of point **A** is nothing but the average neighborhood density(lrd) of point **A** divided by the density of **A**

Interpretation of LOF

- If **LOF(A) = 1**, then we can say that the point has the same density(lrd) as its **k** nearest neighbors
- If **LOF(A) > 1**, then the **k** neighbors of point **A** have a higher density than point **A**.
  - That does not mean point **A** is an outlier. It may or may not be.
  - But if **LOF(A) >>> 1**, then the point is definitely an outlier.
- If **LOF(A) < 1**, then the point has more density than its nearest neighbors.

Disadvantages of LOF

- Finding optimal K
- Finding threshold.
  - If LOF(A) >> 1, what is the threshold??
- Cannot handle high dimensional data efficiently
- High Time Complexity