Basic Methods of Detecting Multidimensional Outliers

S. M. Shakila Arafat Chy
(ID: 277187)
13.12.2023

Outlook

- The concept of outliers
- Reason for seeking outliers
- Importance of detecting outliers
- Overview of methods of detecting univariate outliers
- Why Univariate Analysis is not enough
- Basics methods of detecting multidimensional outliers
- Conclusion
- References

The concept of Outlier/Anomalies

- Outlier is a data point in the dataset that differs **significantly** from the other data or observations. Just look at the picture on the right, there is a series of apples, but one is colored differently. This one apple is what we called an outlier.

- **Noise** and **Anomalies** are related but distinct concepts.
- **Outlier** is part of the data, but **Noise** is just a random error or variance (could be mislabeled or mistake or even missing data).

Importance of detecting outliers

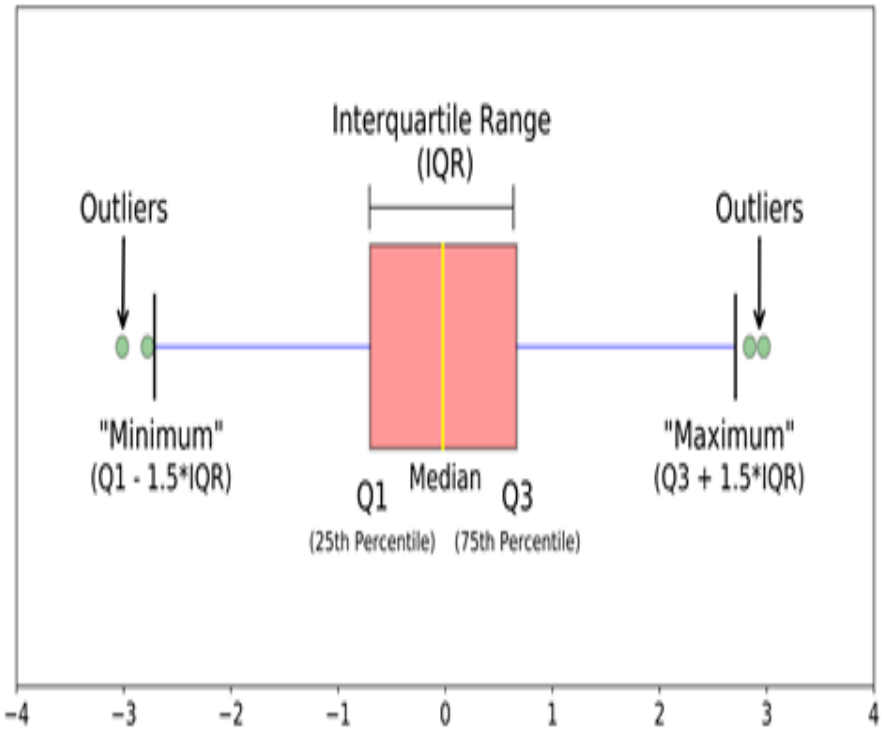
Presence of outliers may cause problems during model fitting (esp. linear models) and may also result in inflated error metrics which give higher weights to large errors. Hence, it is necessary to treat outliers before building a machine learning model.
A more exhaustive list of applications that utilize outlier detection is:

- **Fraud detection** - detecting fraudulent applications for credit cards, state benefits or detecting fraudulent usage of mobile phones.
- **Intrusion detection** - detecting unauthorised access in computer networks.
- **Fault diagnosis** - monitoring processes to detect faults in motors, generators, pipelines or space instruments.
- **Structural defect detection** - monitoring manufacturing lines to detect faulty production runs for example cracked beams.
- **Satellite image analysis** - identifying novel features or misclassified features
- **Medical condition monitoring** - such as heart-rate monitors etc.

Overview of methods of detecting univariate outliers

A univariate outlier is a data point that consists of an extreme value on one variable. Univariate outliers are 1D outliers. Some of the techniques of detecting univariate outliers are:

- Boxplots
- Z-score
- Inter Quantile Range(IQR)



Using the Inter Quantile Range(IQR):

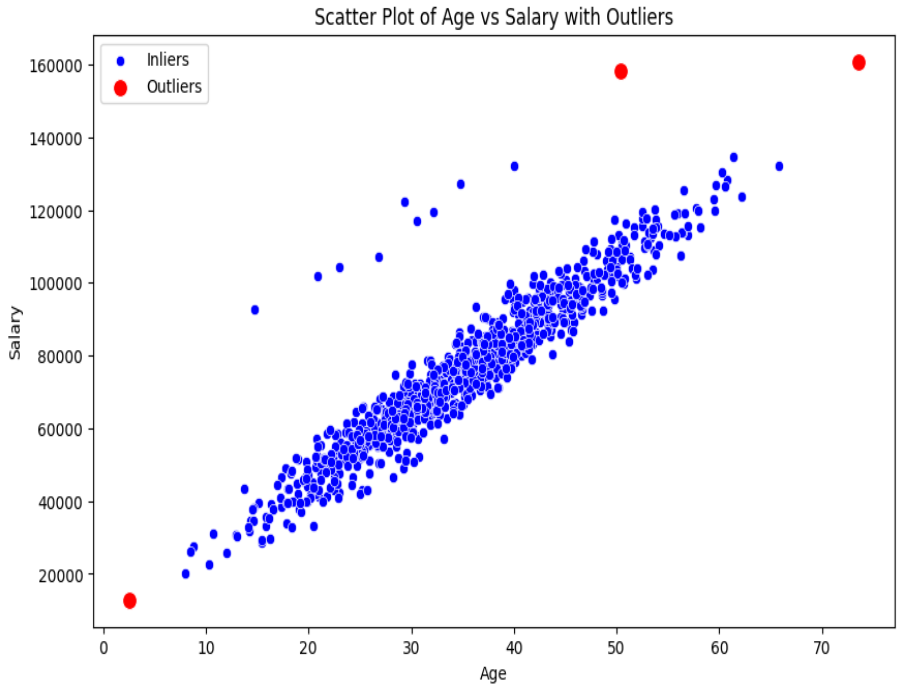
- Sort the dataset in ascending order
- calculate the 1st and 3rd quartiles(Q1, Q3)
- compute IQR=Q3-Q1 (where IQR ≡ Interquartile Range).
- compute lower bound = (Q1–1.5*IQR), upper bound = (Q3+1.5*IQR)
- loop through the values of the dataset and check for those who fall below the lower bound and above the upper bound and mark them as outliers

Multidimensional outlier

- A multivariate outlier is an unusual combination of data points or values in an observation across several variables. Multivariate outliers are multidimentional outliers. In a single variable, the data might not be an outlier, but when it is associated with another variable, the outlier might occur. This is what we call multidimensional outlier.

- This graph illustrates a scatter plot representing the relationship between salary and age variables. "The blue points on the scatter plot form a noticeable pattern, indicating a general association between age and salary variables.“

- However, the plot reveals the presence of outliers, highlighted in red. These red points stand out as outliers, signifying a significant deviation from the overall trend observed in the data

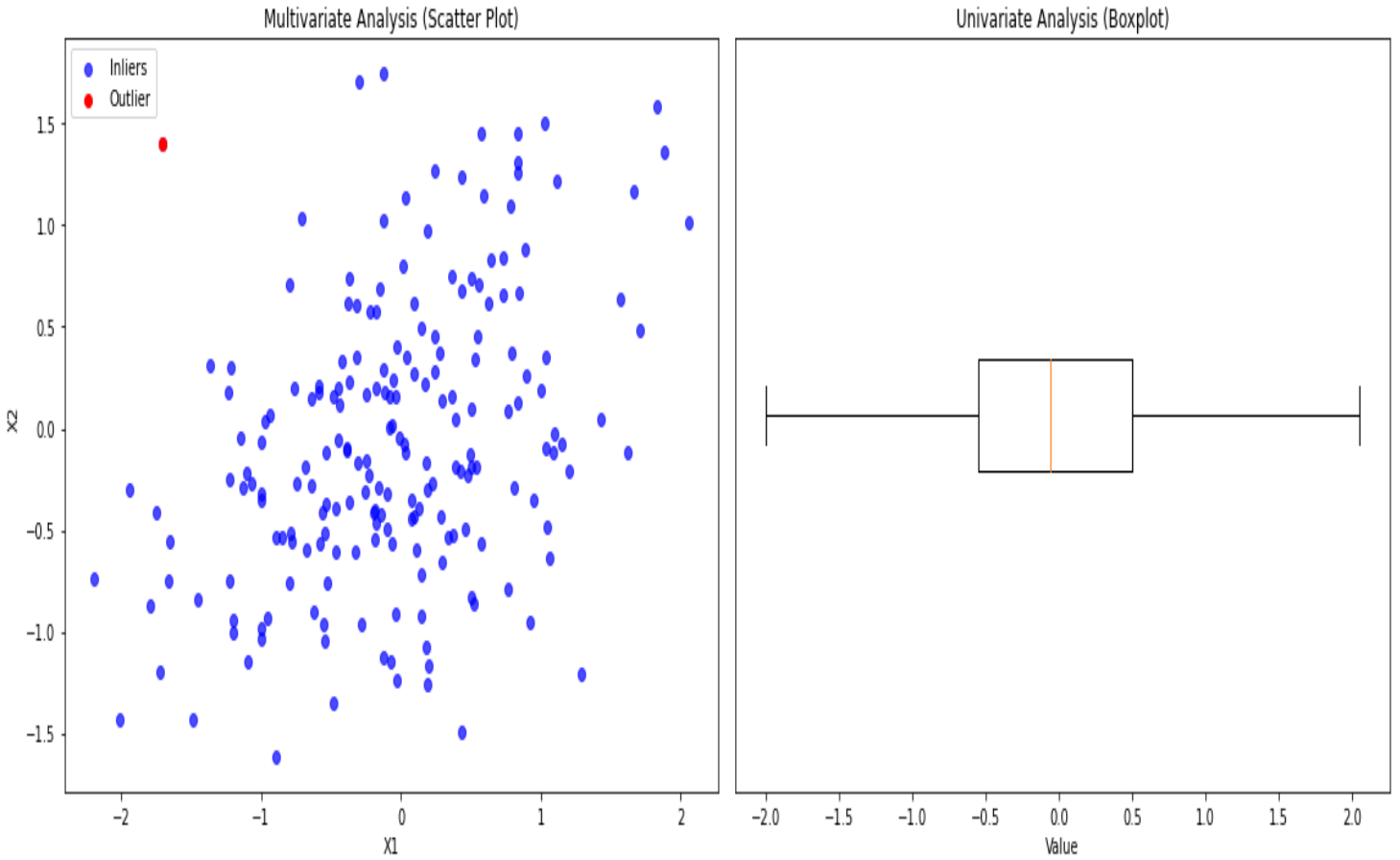


Univariate Analysis is not sufficient

- Let's compare the multivariate analysis (scatter plot) with the univariate analysis (boxplot) to showcase the limitations of univariate analysis in identifying outliers.

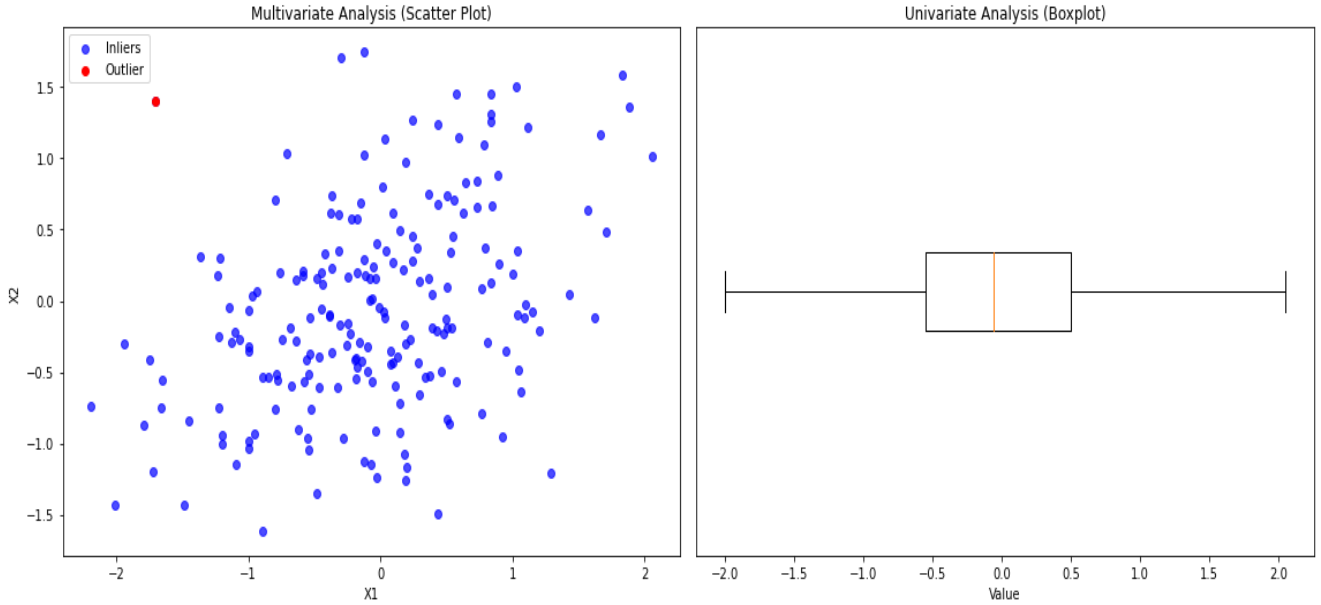
- **Multivariate Analysis (Scatter Plot):**

- The left subplot shows a scatter plot in two dimensions (X1 and X2) for a dataset that includes both inliers (normal data points) and one outlier. Inliers are represented by blue points, and the outlier is represented by a red point. The scatter plot allows us to visualize the entire data distribution and the position of the outlier in the context of both variables



Univariate Analysis is not sufficient

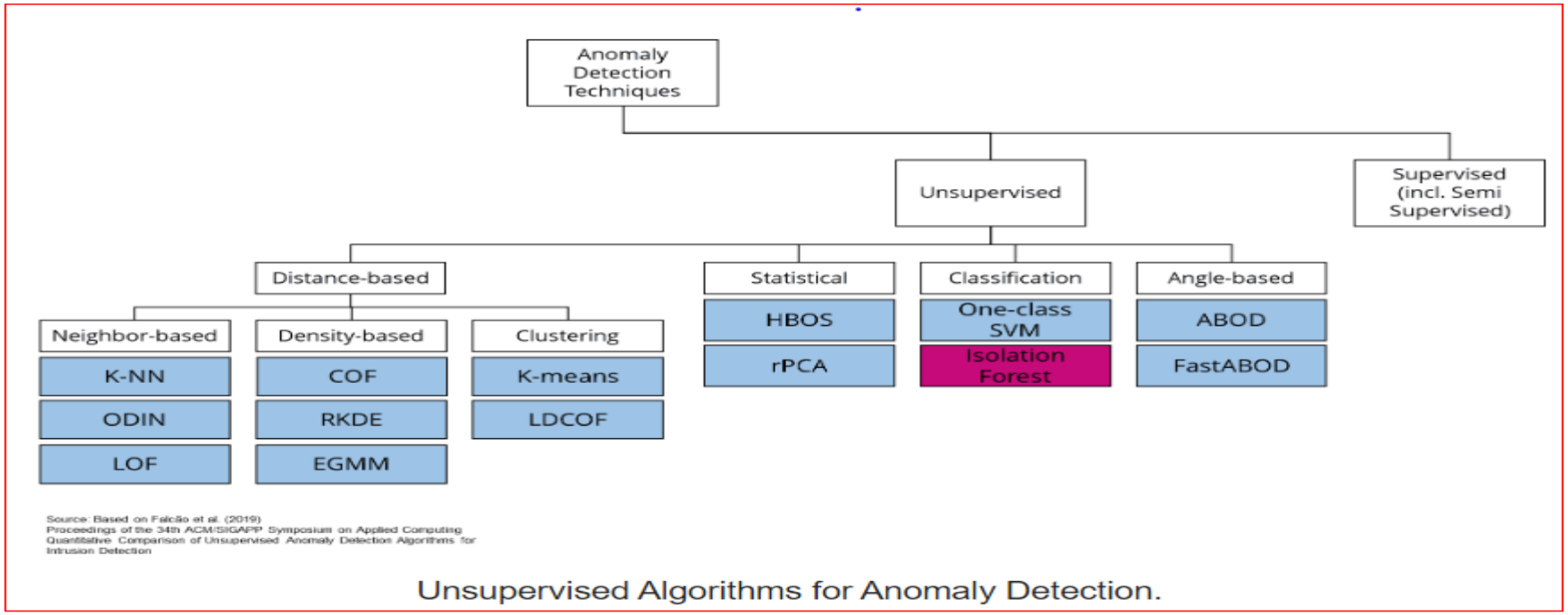
Univariate Analysis (Boxplot):
The right subplot shows a boxplot for one dimension (X1) of the dataset. The boxplot is a univariate representation that summarizes the distribution of values in X1. However, the boxplot does not provide information about the relationship between X1 and X2.



Comparison and Interpretation: The multivariate scatter plot provides a comprehensive view of the data distribution, making it easy to identify the outlier in the context of both variables. The univariate boxplot, on the other hand, may not effectively capture the outlier because it only considers the distribution of values along a single dimension (X1). This comparison illustrates the importance of multivariate analysis in identifying outliers, as univariate methods may overlook anomalies that are evident in a broader context.

Anomaly Detection Techniques

- Outlier detection is a classification problem. In this field, we use both supervised (where the algorithm learns from labeled examples) and unsupervised (where the algorithm figures it out on its own) machine learning techniques. The chart below gives a clear picture of the common algorithms used for unsupervised outlier detection.



Unsupervised Algorithms for Anomaly Detection.

Basic methods of detecting multidimensional outliers

Isolation Forest(iForest)

- Isolation forests are known to work well for high dimensional data. Isolation Forest, like any tree ensemble method, is built based on decision trees.
- An important concept in this method is the isolation number.
- The isolation number is the number of splits needed to isolate a data point.

iForest

Here is briefly how Isolation forests work:

- Construct an Isolation Tree either from the entire feature set or a randomly chosen subset of the feature set.
- Construct **n** such Isolation trees.
- Calculate an Anomaly score for each data point. **The Anomaly score is a non-linear function of the Average path length over all Isolation trees.**
- An anomaly score s(x) is defined as :

$$s(x) = 2^{\frac{-h(x)}{E(h(x)}} \;;$$

where $h(x)$ is the path length of instance x in the tree

$E\big(h(x)\big)$Is the average path length in the tree.

- Instances with scores closer to 1 are more likely to be anomalies.
- The path length is equivalent to the number of splits made by the Isolation tree to isolate a point. The shorter the Average path length, the larger are the chances of the point being an anomaly

iForest

It has 2 Important methods:

Decision_function(X): Returns a score — such that examples having more negative scores are more anomalous.

Predict(X): Returns -1 for Anomalous points and +1 for normal points. The number of points output as anomalous depends on the contamination value set while fitting the model.

iForest

The Isolation forest in **skearn** has 4 important inputs:

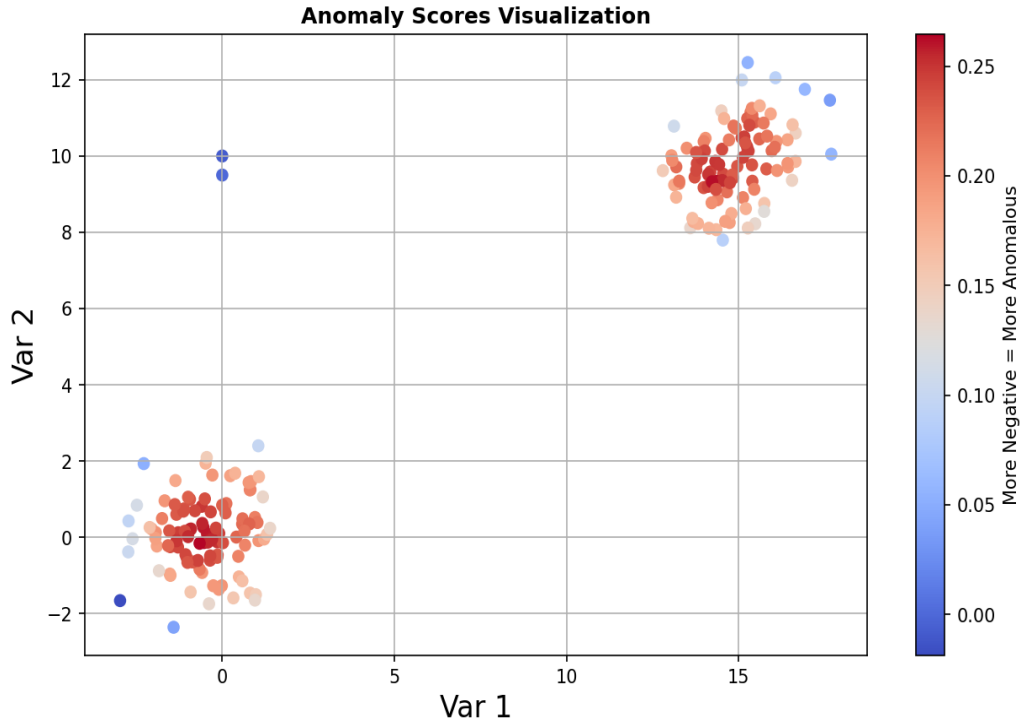
- **n_estimators:** Number of Isolation trees trained.
- **max_samples:** Number of data points used to train each tree.
- **contamination:** Fraction of anomalous data points. For example, if we suspect 5% of the data to be anomalous, we set contamination to 0.05
- **max_features:** Number of features to be used to train each tree(This is in contrast to Random Forests where we decide on a random subset of features for each split*).*

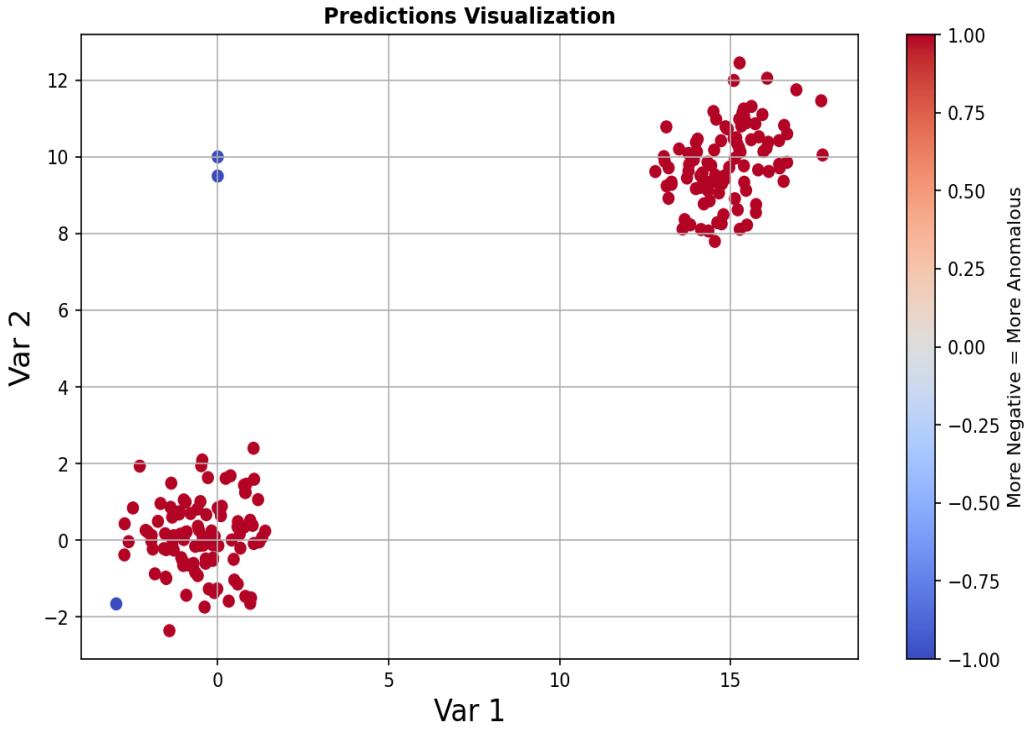
iForest

This code generates artificial data with multivariate outliers, trains an Isolation Forest model on the data, and then visualizes the anomaly scores and predictions in two separate plots.

```python
# Create Artificial Data with Multivariate Outliers
d1 = np.random.multivariate_normal(mean=np.array([-.5, 0]),
                                   cov=np.array([[1, 0], [0, 1]]), size=100)
d2 = np.random.multivariate_normal(mean=np.array([15, 10]),
                                   cov=np.array([[1, 0.3], [0.3, 1]]), size=100)
outliers = np.array([[0, 10], [0, 9.5]])
d = pd.DataFrame(np.concatenate([d1, d2, outliers], axis=0), columns=['Var 1', 'Var 2'])

# Train Isolation Forest
model = ensemble.IsolationForest(n_estimators=50, max_samples=500, contamination=.01, max_features=2,
                                 bootstrap=False, n_jobs=1, random_state=1, verbose=0, warm_start=False).fit(d)

# Get Anomaly Scores and Predictions
anomaly_score = model.decision_function(d)
predictions = model.predict(d)

# Visualize Anomaly scores
plt.figure(figsize=(10, 6), dpi=150)
s = plt.scatter(d['Var 1'], d['Var 2'], c=anomaly_score, cmap='coolwarm')
plt.colorbar(s, label='More Negative = More Anomalous')
plt.xlabel('Var 1', fontsize=16)
plt.ylabel('Var 2', fontsize=16)
plt.grid()
plt.title('Anomaly Scores Visualization', weight='bold')
plt.show()

# Visualize Predictions
plt.figure(figsize=(10, 6), dpi=150)
s = plt.scatter(d['Var 1'], d['Var 2'], c=predictions, cmap='coolwarm')
plt.colorbar(s, label='More Negative = More Anomalous')
plt.xlabel('Var 1', fontsize=16)
plt.ylabel('Var 2', fontsize=16)
plt.grid()
plt.title('Predictions Visualization', weight='bold')
plt.show()
```

iForest





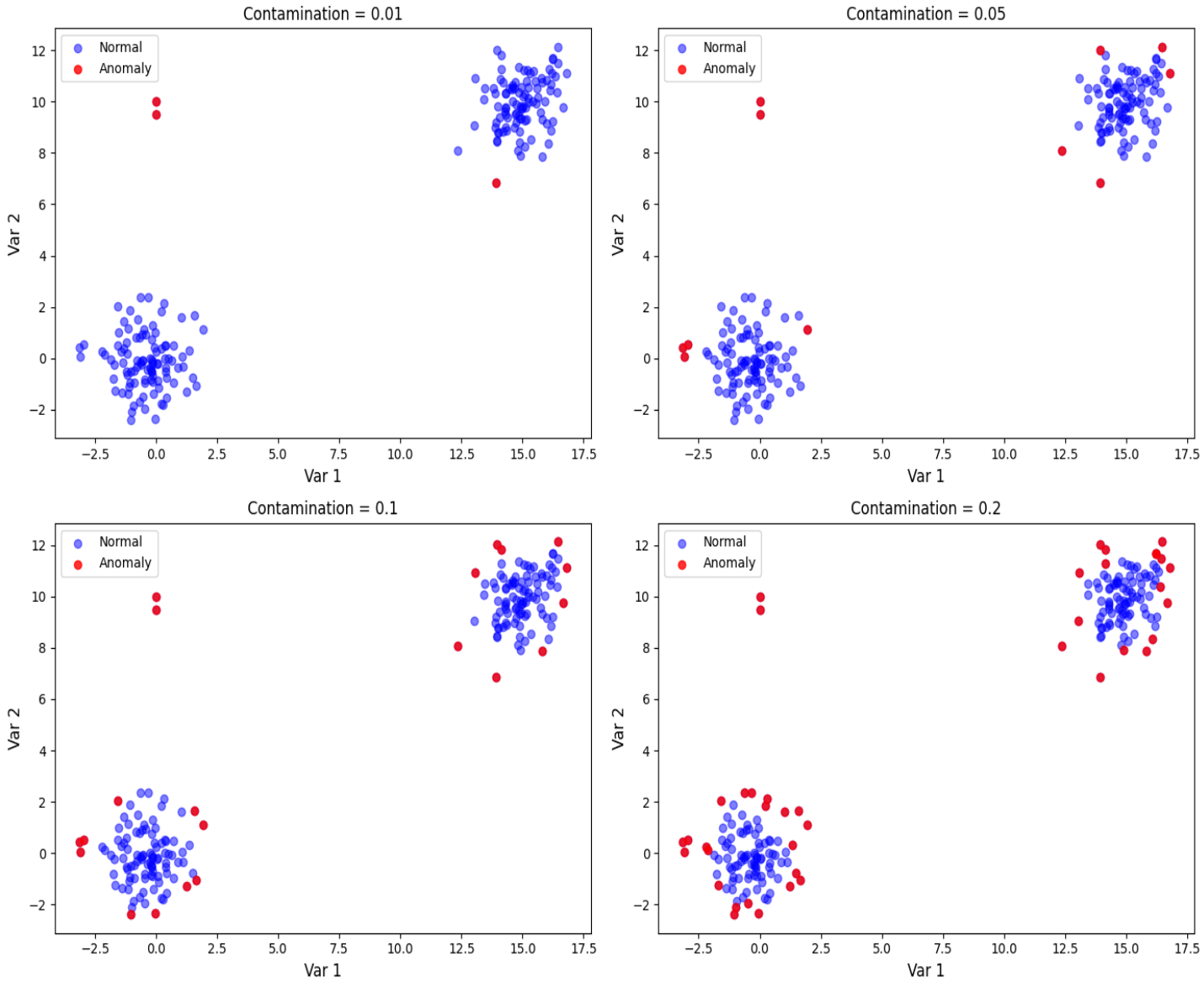
Anomaly Scores Visualization:
The first visualization displays anomaly scores for each data point.
Anomaly scores quantify the degree of abnormality, with more negative scores indicating higher anomaly.

Predictions Visualization:
The second visualization shows predictions from the Isolation Forest model.
Predictions are binary (1 for inliers, -1 for outliers) and aligned with anomaly scores.

iForest

With increasing Contamination values, the model labels more data as anomalous.

Lastly,To know the right contamination we can analyze the decision_function() output.



Strengths and weakness of iForest

Strength	Weakness
It is fast, accurate	It is weak in dealing with local outliers
Work well with large dataset	
No need to calculate distance, hence save memory	

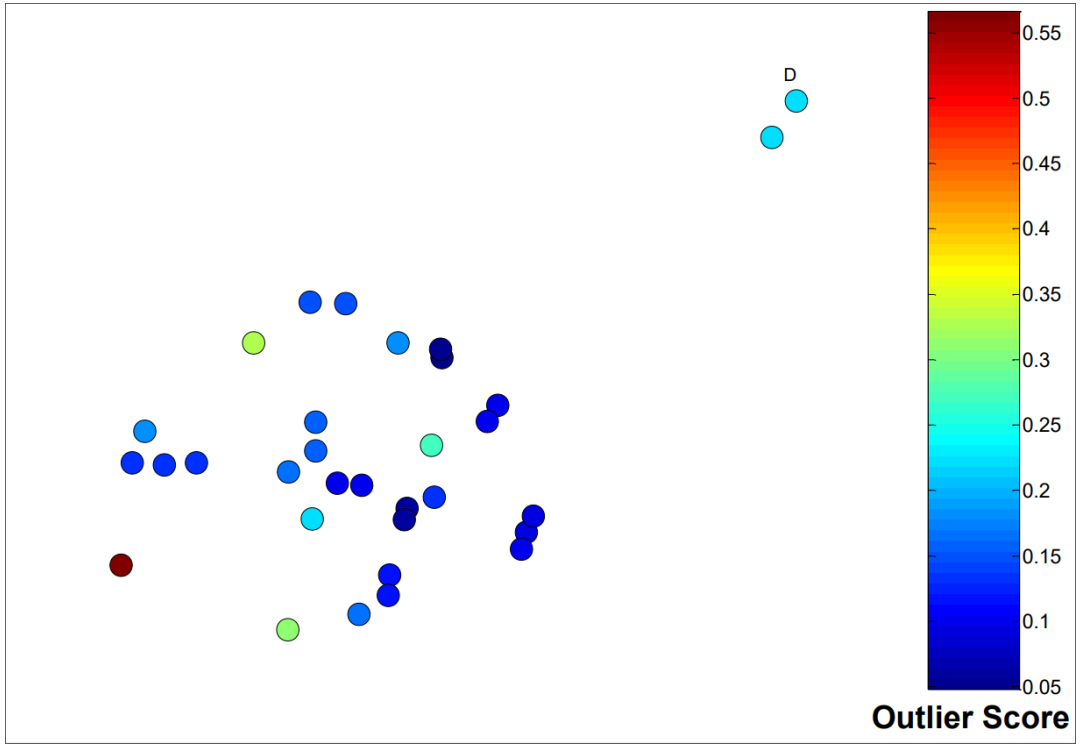
Distance to k-th nearest neighbor

- The K-Nearest Neighbour (KNN) algorithm is a non-parametric method used for classification, regression and anomaly detection.

- It is based on feature similarity , which means it classifies a new object based on how closely it resembles the objects in the training set.

Distance to k-th nearest neighbor (one NN)

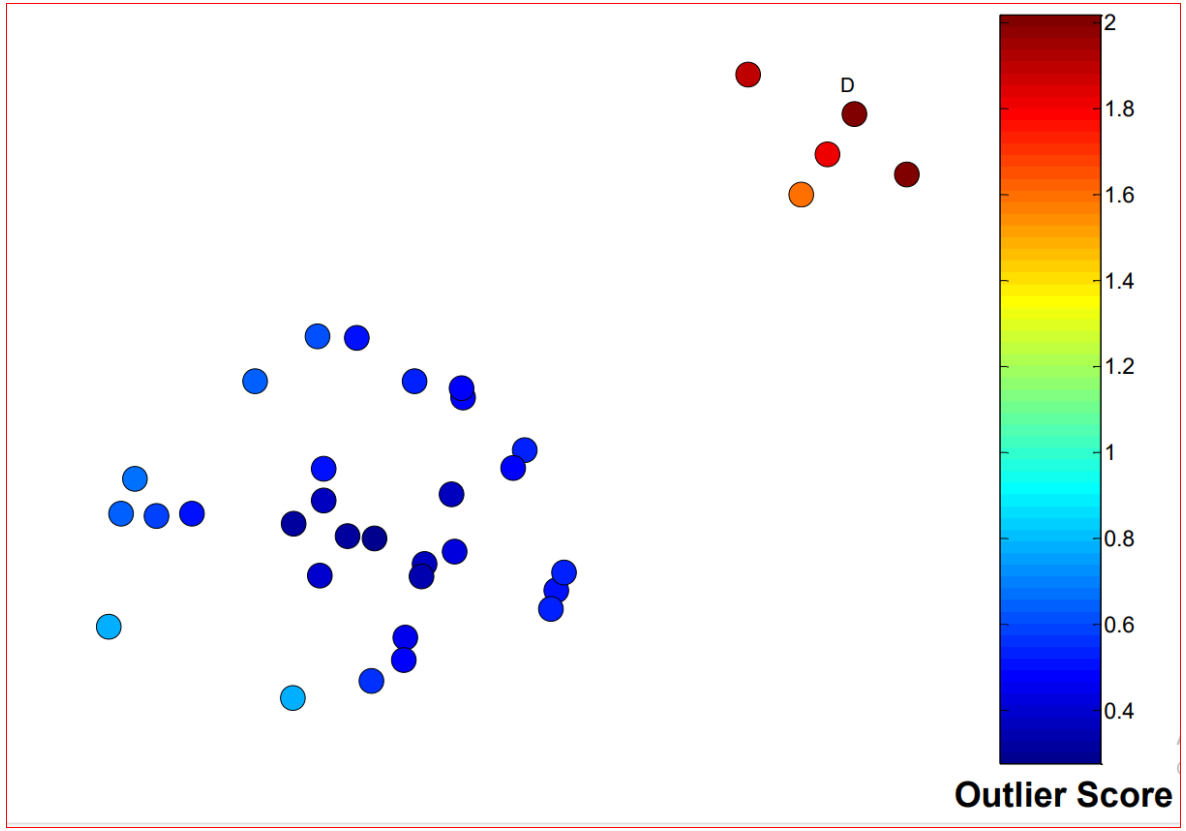
- The k-NN method is sensitive to how it starts. Initial choices significantly impact how it groups items.
- In the same way that clustering methods can be affected by initial settings, the k-NN method is also sensitive to its starting conditions This sensitivity can lead to the misclassification of outlier clusters."



Source: [TSKK] chapter 9.

Distance to k-th nearest neighbor

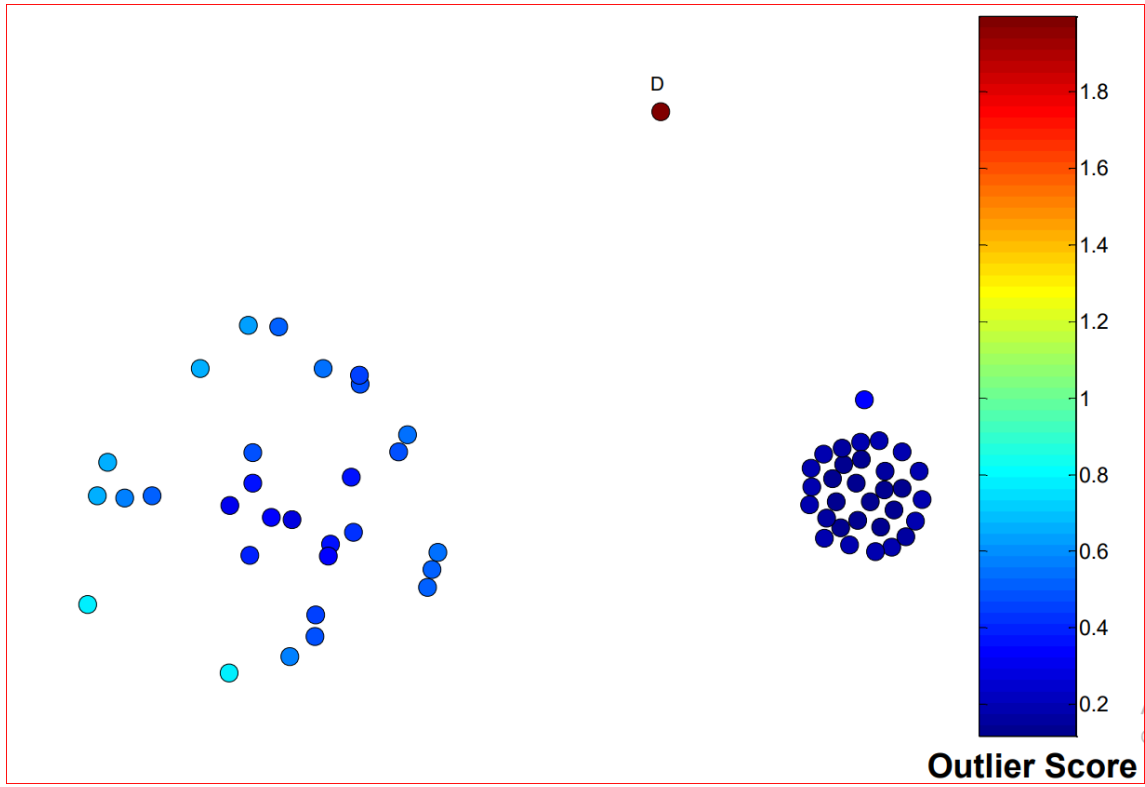
- (Five nearest neighbour : small cluster)
- For big enough k parameter, some clusters may be classified as outliers, where in reality they could be a part of good data



[TSKK] chapter 9.

Distance to k-th nearest neighbor(Differing Density)

- k-NN method by itself doesn't take into account the density of the clusters, there is a point close to the smaller cluster that could be marked as an outlier



[TSKK] chapter 9.

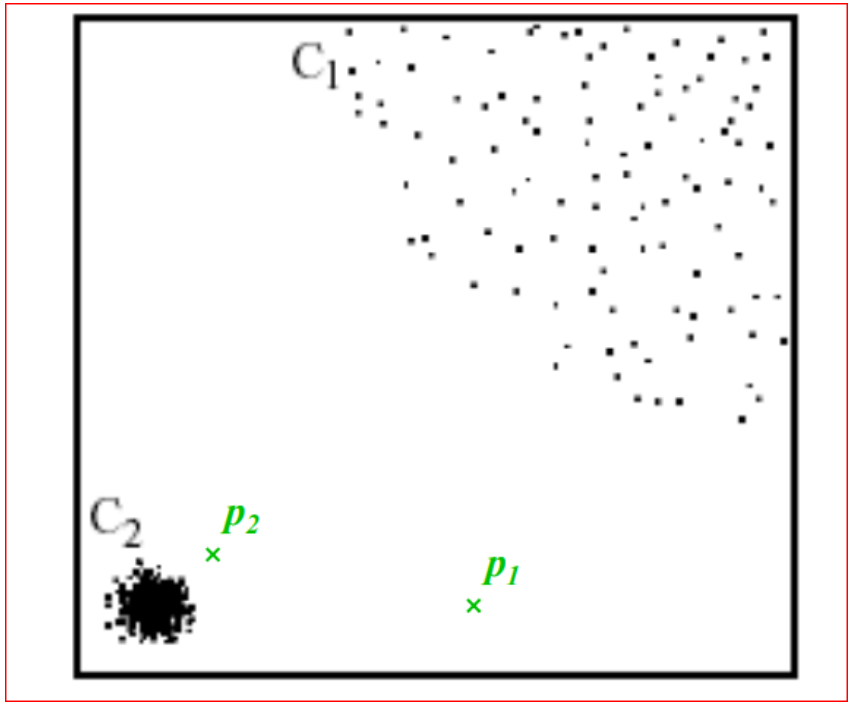
Relative density - local outlier factor (LOF) approach

- **Density:** We think of density as the opposite of the average distance to the nearest neighbors (closer neighbors imply higher density).
- **Relative Density:** For a point *X*, we calculate its relative density by comparing its density to the average density of its *k*-nearest neighbors.

- **LOF Approach:** The Local Outlier Factor (LOF) method computes an outlier score based on this relative density. Simply put, it identifies outliers when a point has lower local density compared to its neighbors. In other words, it spots anomalies by looking for points in regions that are less crowded compared to their surroundings.

Relative Density-based: LOF approach

- For each point, compute the density of its local neighborhood
- Compute local outlier factor (LOF) of a sample p as the average of the ratios of the density of sample p and the density of its nearest neighbors
- Outliers are points with largest LOF value
- In the NN approach, p2 is not considered as outlier, while LOF approach find both p1 and p2 as outliers



Strengths/Weaknesses of Density-Based Approaches

Strengths:

- Ability to Detect Arbitrary-Shaped Clusters
- Parameter Tuning Flexibility:
- Applicability to High-Dimensional Data
- Noise Handling

Weakness:

- Expensive –can become computationally expensive, especially as the dataset size (n) grows
- Sensitive to parameters
- Density becomes less meaningful in highdimensional spac

References.

- https://towardsdatascience.com/anomaly-detection-in-python-part-2-multivariate-unsupervised-methods-and-code-b311a63f298b
- https://towardsdatascience.com/multi-variate-outlier-detection-in-python-e900a338da10
- https://cxl.com/blog/outliers/
- https://www-users.cse.umn.edu/~kumar001/dmbook/slides/chap9_anomaly_detection.pdf
- TSKK(Chapter 9)