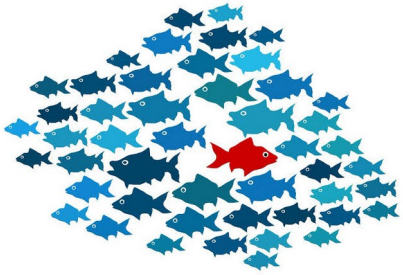
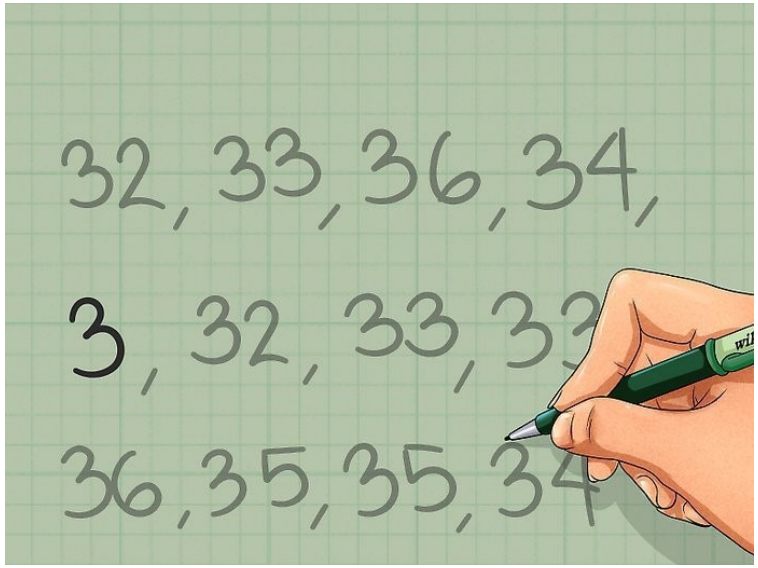
Outliers or Anomaly Detection

Tushar B. Kute,
http://tusharkute.com

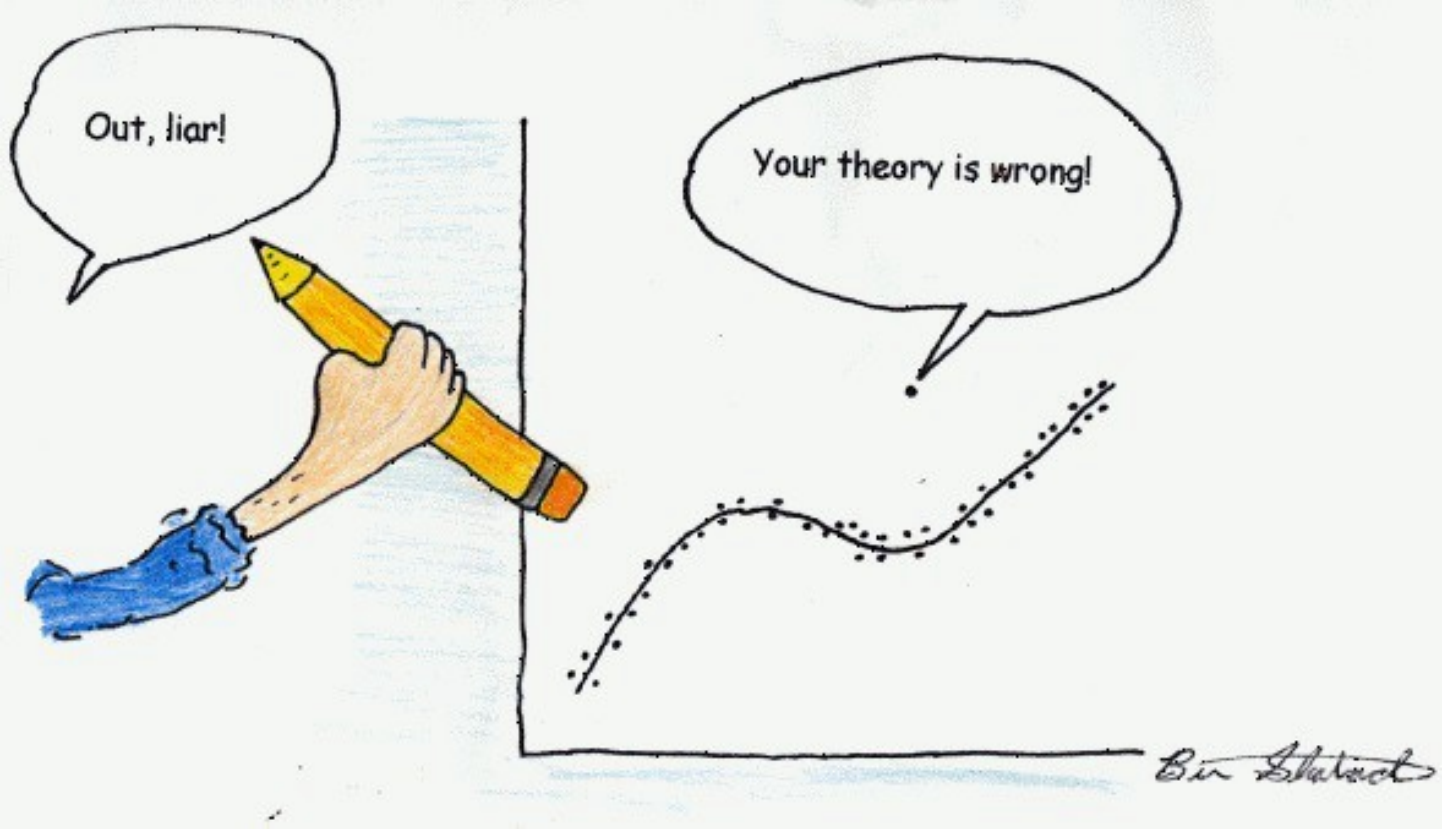


Outlier ?

- In statistics, an outlier is an observation point that is distant from other observations.
- The above definition suggests that outlier is something which is separate/different from the crowd.



Outlier when plotted



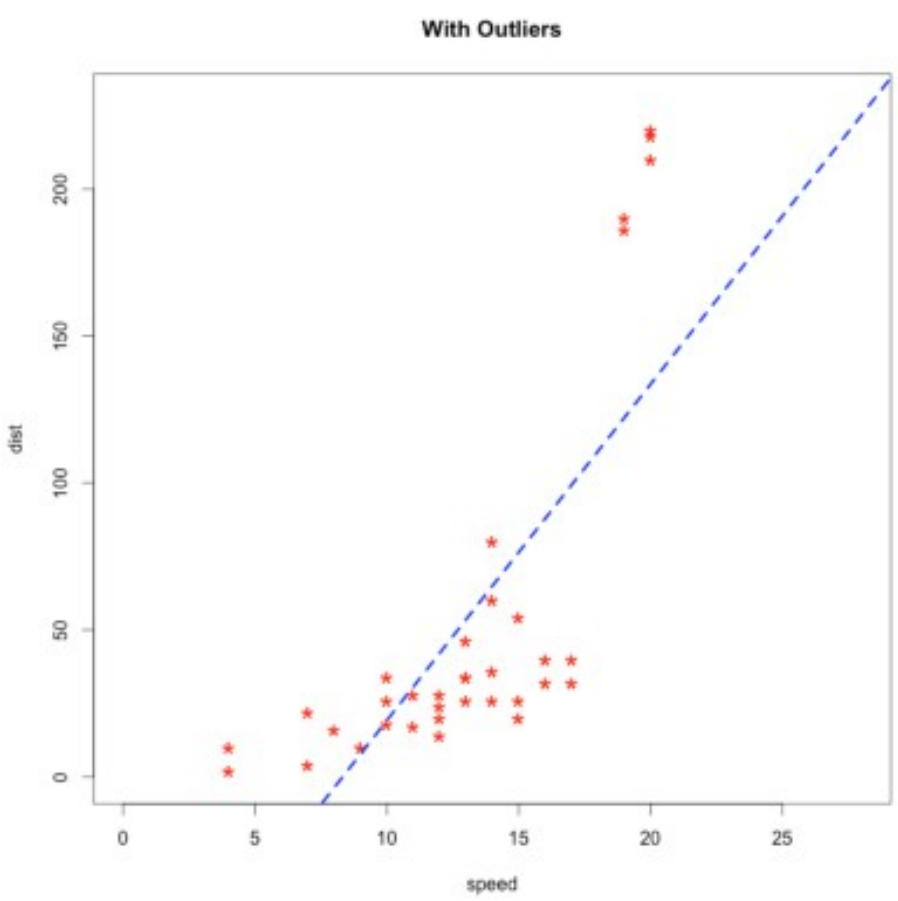
See through examples

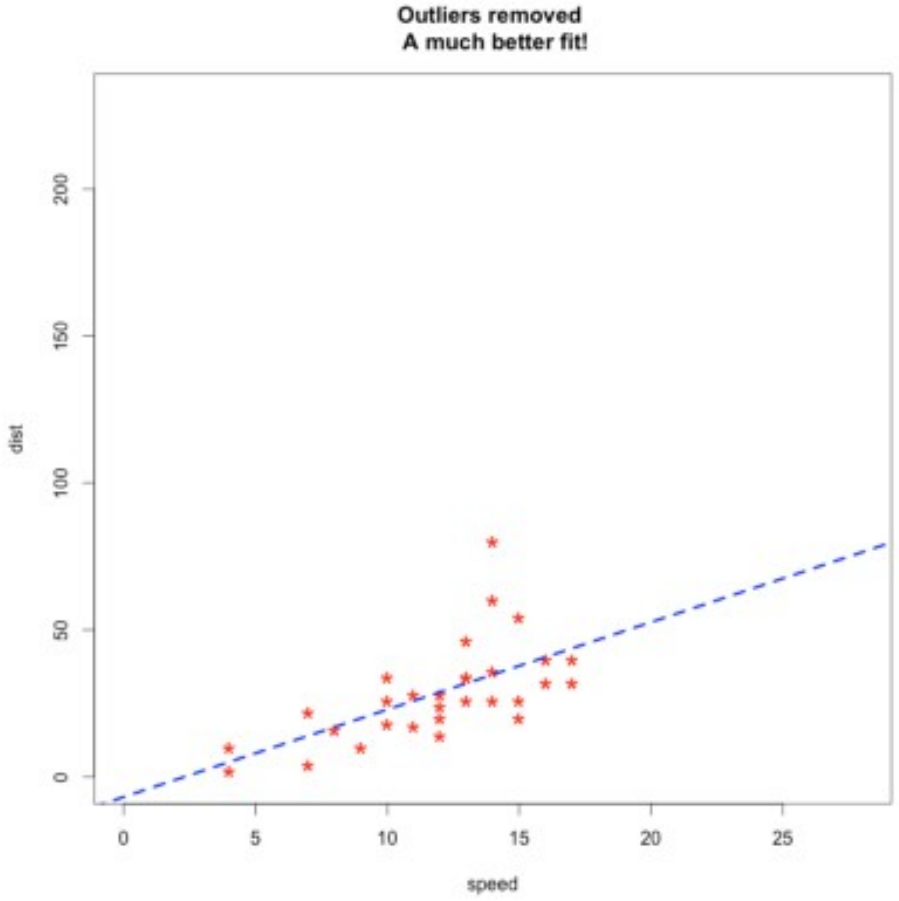
- An outlier is any data point which differs greatly from the rest of the observations in a dataset. Let's see some real life examples to understand outlier detection:
  - When one student averages over 90% while the rest of the class is at 70% – a clear outlier
  - While analyzing a certain customer's purchase patterns, it turns out there's suddenly an entry for a very high value. While most of his/her transactions fall below Rs. 10,000, this entry is for Rs. 1,00,000. It could be an electronic item purchase – whatever the reason, it's an outlier in the overall data
  - How about Usain Bolt? Those record breaking sprints are definitely outliers when you factor in the majority of athletes.

Outlier Types

- Outliers are of two types: Univariate and Multivariate.
- A univariate outlier is a data point that consists of extreme values in one variable only, whereas a multivariate outlier is a combined unusual score on at least two variables.
- Suppose you have three different variables – X, Y, Z. If you plot a graph of these in a 3-D space, they should form a sort of cloud.
- All the data points that lie outside this cloud will be the multivariate outliers.

Why to detect outliers?





What did they say ?

- *"Outliers are not necessarily a bad thing. These are just observations that are not following the same pattern as the other ones. But it can be the case that an outlier is very interesting. For example, if in a biological experiment, a rat is not dead whereas all others are, then it would be very interesting to understand why. This could lead to new scientific discoveries. So, it is important to detect outliers."*

  – Pierre Lafaye de Micheaux, Author and Statistician

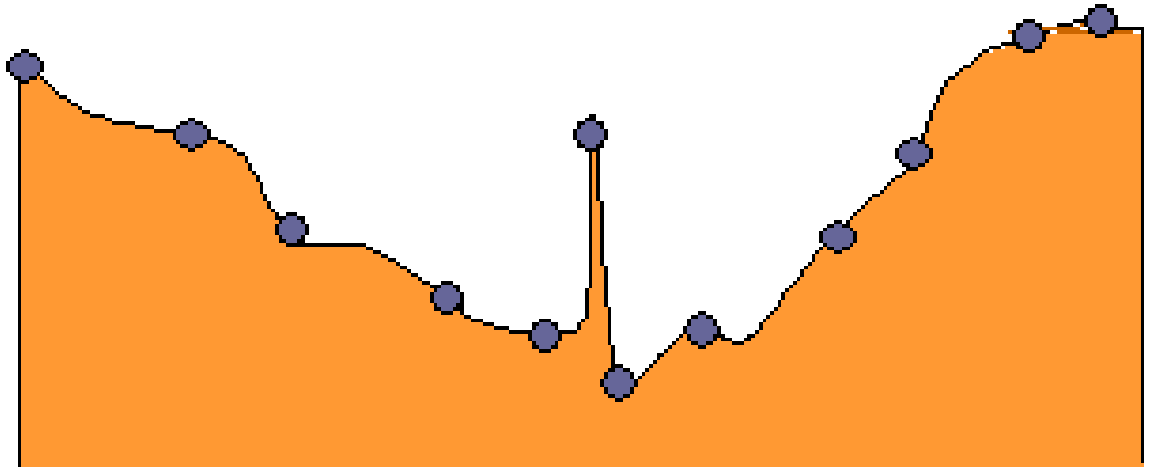
Prime Applications

- Fault diagnosis,
- Intrusion detection
- Fraud Detection
- Web analytics
- Medical diagnosis
- Financial industry
- Quality control

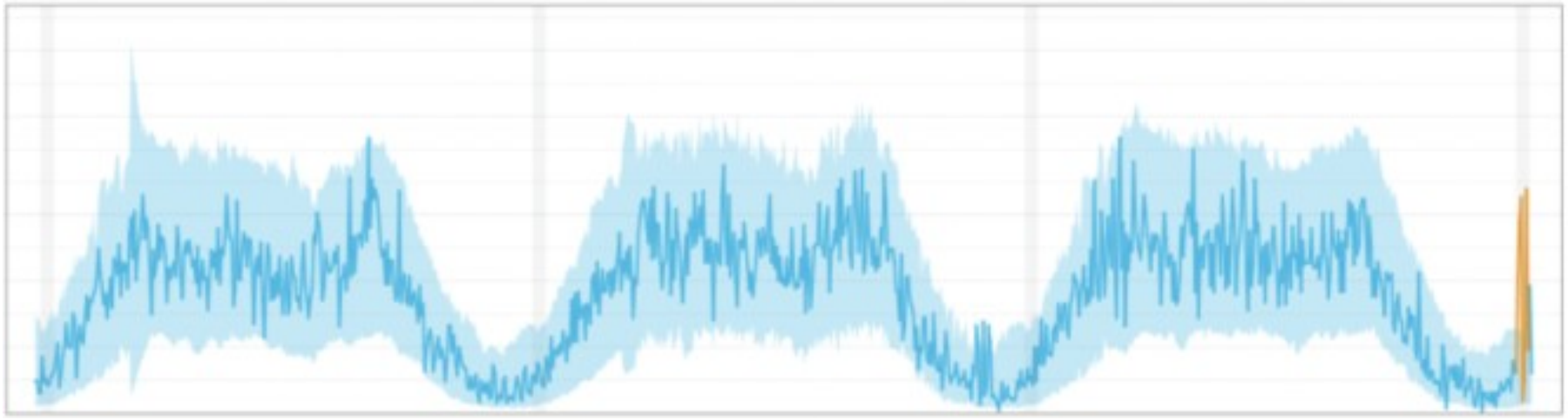
Types of Anomalies

- Global Anomalies
- Contextual Anomalies
- Collective Anomalies

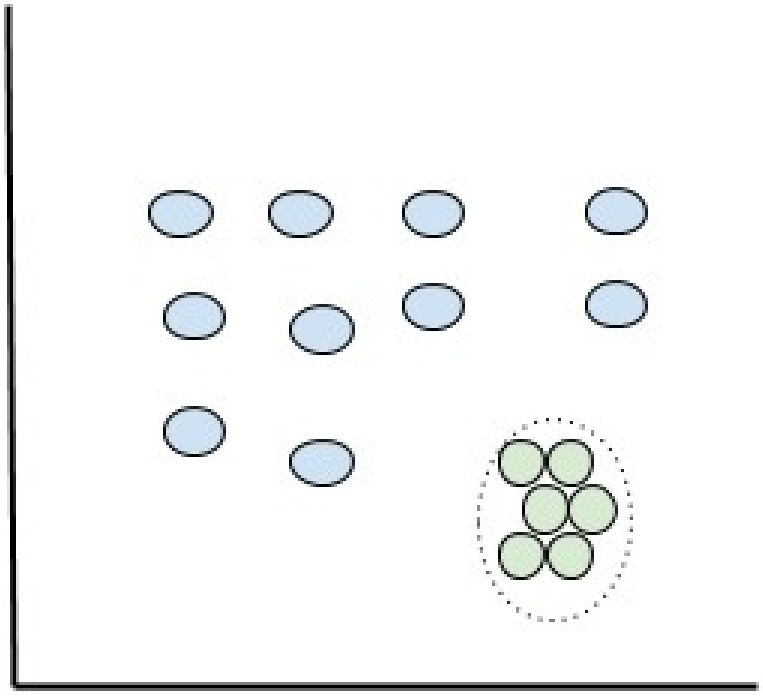
Global Anomalies



Contextual Anomalies



Collective Anomalies



General Methods for Detection

- Box Plot
- Histogram
- Clustering
- Isolation Forest

Packags needed

- Data Analytics:
  - pandas
- Numerical Python:
  - Numpy
  - scipy
- Random Number
  - faker
- Visualization
  - Matplotlib

Let's Start

```
# Import the necessary packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Use a predefined style set
plt.style.use('ggplot')

# Import Faker
from faker import Faker
fake = Faker()

# To ensure the results are reproducible
fake.seed(4321)
names_list = []
```

Create a random list

```
for _ in range(100):
  names_list.append(fake.name())

# To ensure the results are reproducible
np.random.seed(7)

salaries = []
for _ in range(100):
    salary = np.random.randint(1000,2500)
    salaries.append(salary)

# Create pandas DataFrame
salary_df = pd.DataFrame({'Person': names_list,
     'Salary': salaries  })
```

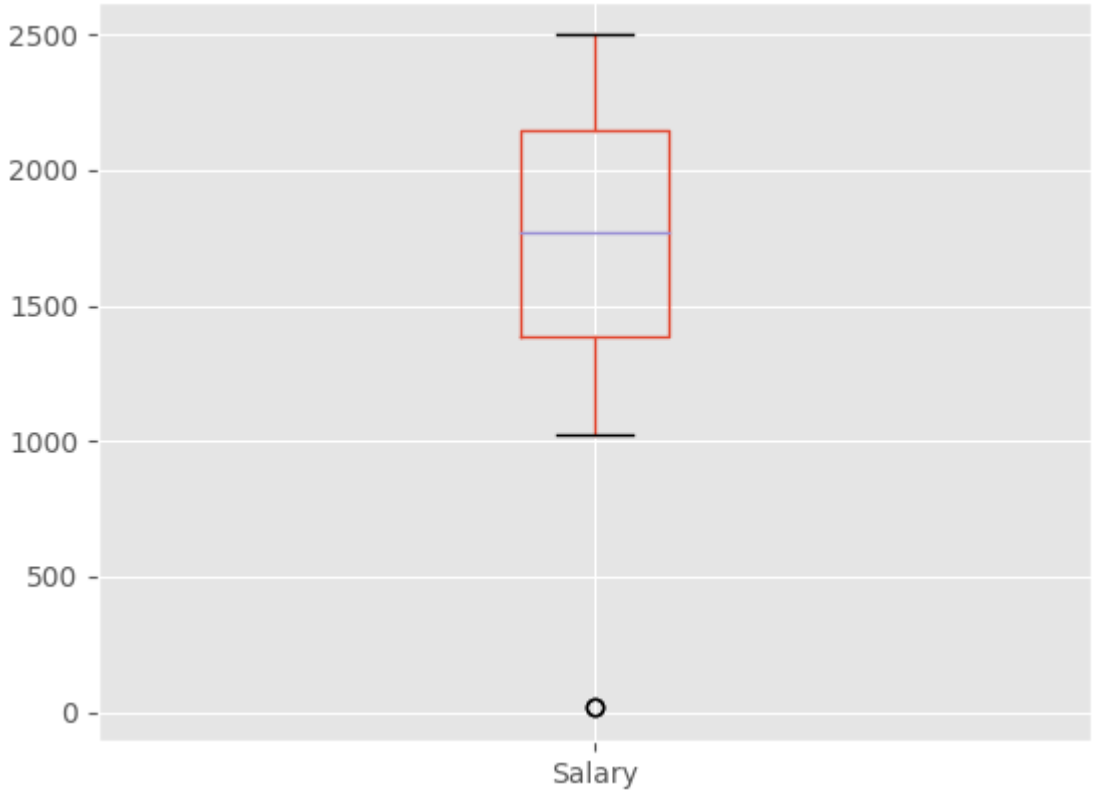
Add outliers and view

```
# Print a subsection of the DataFrame
print(salary_df.head())

salary_df.at[16, 'Salary'] = 23
salary_df.at[65, 'Salary'] = 17

# Verify if the salaries were changed
print(salary_df.loc[16])
print(salary_df.loc[65])

# Generate a Boxplot
salary_df['Salary'].plot(kind='box')
plt.show()
```

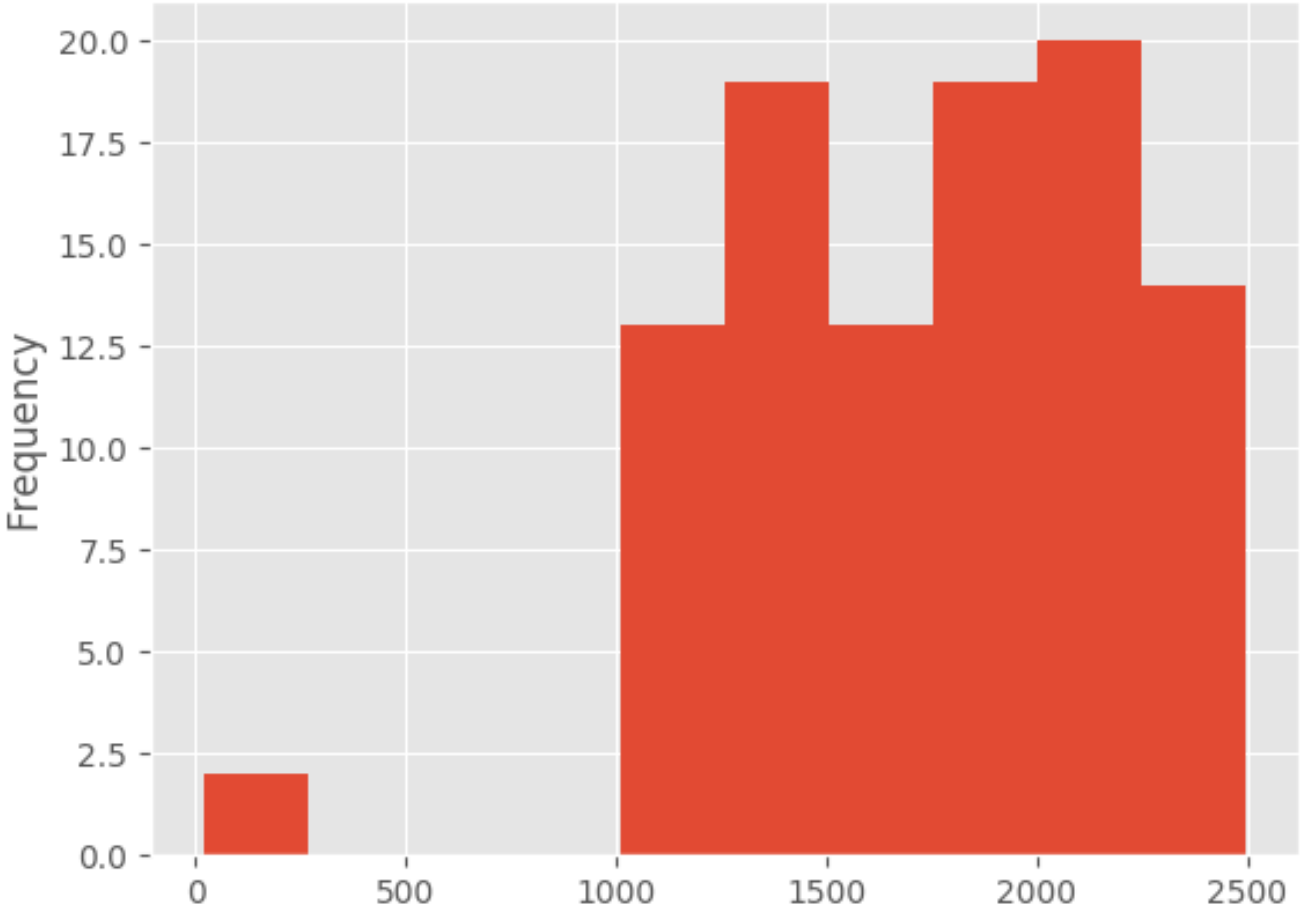
Check the outliers

```
# Generate a Boxplot
salary_df['Salary'].plot(kind='box')
plt.show()

# Generate a Histogram plot
salary_df['Salary'].plot(kind='hist')
plt.show()

# Minimum and maximum salaries
print('Min salary ' + str(salary_df['Salary'].min()))
print('Max salary ' + str(salary_df['Salary'].max()))
```

Boxplot



Histogram


Frequency
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
500
1000
1500
2000
2500

Using Clustering

- We are going to use K-Means clustering which will help us cluster the data points (salary values in our case).
- The implementation that we are going to be using for KMeans uses Euclidean distance internally. Let's get started.

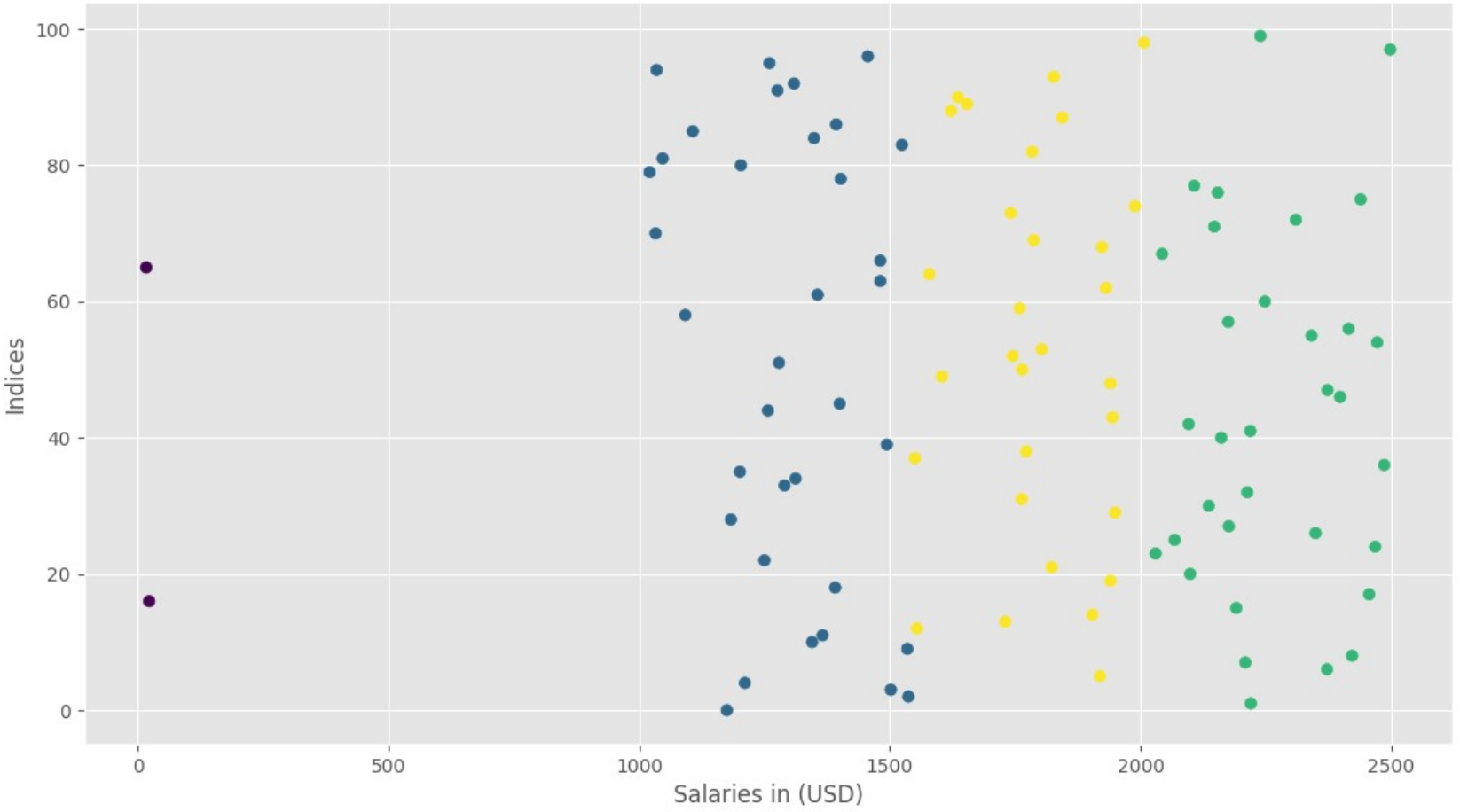
Getting Started

```
# Convert the salary values to a numpy array
salary_raw = salary_df['Salary'].values

# For compatibility with the SciPy
salary_raw = salary_raw.reshape(-1, 1)
salary_raw = salary_raw.astype('float64')

# Import kmeans from SciPy
from scipy.cluster.vq import kmeans
import scipy.cluster as cluster

# Specify the data, the no. of clusters
centroids, avg_distance = kmeans(salary_raw, 4)
```

Create the group and plot

```
# Get the groups (clusters) and distances
groups, cdist = cluster.vq.vq(salary_raw, centroids)

plt.scatter(salary_raw, np.arange(0,100), c=groups)
plt.xlabel('Salaries in (USD)')
plt.ylabel('Indices')
plt.show()
```

Outputs


100
80
60
40
20
0
Indices
0
500
1000
1500
2000
2500
Salaries in (USD)

Automatic Outlier Detection

- Automatic Outlier Detection
  - Isolation Forest
  - Minimum Covariance Determinant
  - Local Outlier Factor
  - One-Class SVM

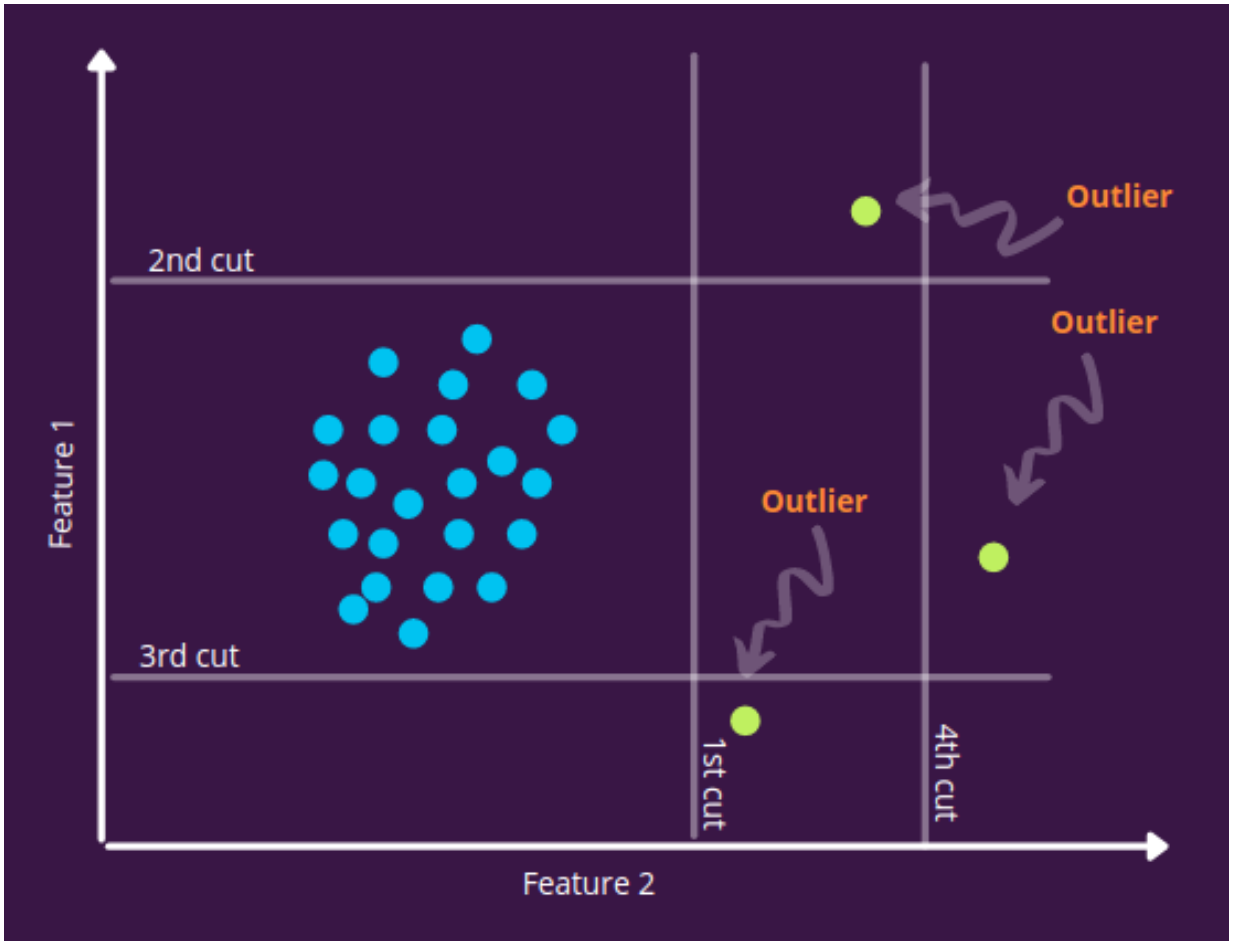
Isolation Forests

- Isolation Forest, or iForest for short, is a tree-based anomaly detection algorithm.
- It is based on modeling the normal data in such a way as to isolate anomalies that are both few in number and different in the feature space.
- This method takes advantage of two anomalies' quantitative properties:
  - i) they are the minority consisting of fewer instances and
  - ii) they have attribute-values that are very different from those of normal instances.

Isolation Forests

- The idea over here is to keep splitting the data at random thresholds and feature till every point gets isolated (it's like overfitting a decision tree on a dataset).
- Once the isolation is achieved we chunk out points that got isolated pretty early during this process.
- And we mark these points as potential outliers. If you see this intuitively, the farther a point is from the majority, the easier it gets to isolate, whereas, isolating the points that are part of a group would require more cuts to isolate every point.

Isolation Forests



Contamination in outliers

- The most important hyperparameter in the model is the “contamination” argument, which is used to help estimate the number of outliers in the dataset.
- This is a value between 0.0 and 0.5 and by default is set to 0.1.

Minimum Covariance Determinant

- If the input variables have a Gaussian distribution, then simple statistical methods can be used to detect outliers.
- For example, if the dataset has two input variables and both are Gaussian, then the feature space forms a multi-dimensional Gaussian and knowledge of this distribution can be used to identify values far from the distribution.

Local Outlier Factor

- A simple approach to identifying outliers is to locate those examples that are far from the other examples in the feature space.
- This can work well for feature spaces with low dimensionality (few features), although it can become less reliable as the number of features is increased, referred to as the curse of dimensionality.

Local Outlier Factor

- The local outlier factor, or LOF for short, is a technique that attempts to harness the idea of nearest neighbors for outlier detection.
- Each example is assigned a scoring of how isolated or how likely it is to be outliers based on the size of its local neighborhood. Those examples with the largest score are more likely to be outliers.
- We introduce a local outlier (LOF) for each object in the dataset, indicating its degree of outlier-ness.

Local Outlier Factor

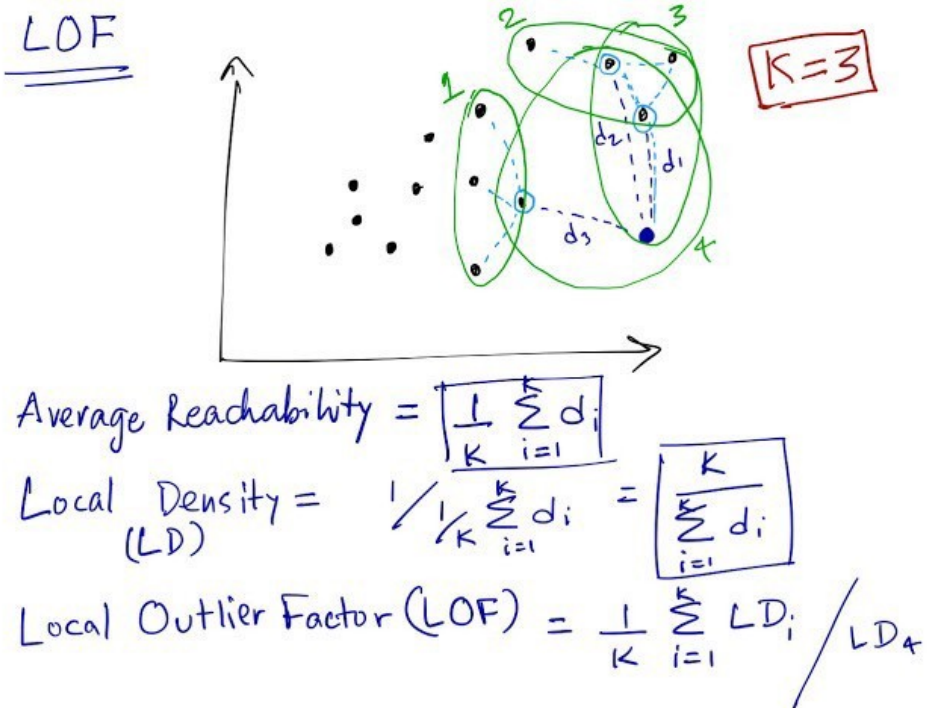
- We calculate and compare the local density of the focus point with the local density of its neighbours.
- If we find that the local density of the focus point is very low compared to its neighbours, that would kind of hint that the focus point is isolated in that space and is a potential outlier.
- The algorithm depends on the hyperparameter K, which decides upon the number of neighbours to consider when calculating the local density.
- This value is bounded between 0 (no neighbour) and the total points (all points being neighbour) in the space.

Local Outlier Factor

- The local density function is defined as the reciprocal of average reachability distance, where, average reachability distance is defined as the average distance from the focus point to all points in the neighbour.
- LOF = average local density of neighbors / local density of focus point
- If,
  - LOF ≈ 1 similar density as neighbors
  - LOF < 1 higher density than neighbors (normal point)
  - LOF > 1 lower density than neighbors (anomaly)

Local Outlier Factor

- The below diagram shows the calculation of LOF and Local Density for a sample focus point(dark blue) in the space. Here, K=3(neighbours), d(distance) can be calculated as euclidean, manhattan, etc.



One Class SVM

- The support vector machine, or SVM, algorithm developed initially for binary classification can be used for one-class classification.
- When modeling one class, the algorithm captures the density of the majority class and classifies examples on the extremes of the density function as outliers.
- This modification of SVM is referred to as One-Class SVM.

One Class SVM

- It is an algorithm that computes a binary function that is supposed to capture regions in input space where the probability density lives (its support), that is, a function such that most of the data will live in the region where the function is nonzero.
- Although SVM is a classification algorithm and One-Class SVM is also a classification algorithm, it can be used to discover outliers in input data for both regression and classification datasets.

Useful resources

- www.scikit-learn.org
- www.towardsdatascience.com
- https://machinelearningmastery.com
- www.medium.com
- www.analyticsvidhya.com
- www.depends-on-the-definition.com
- www.kaggle.com
- www.github.com

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
https://mituresearch.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com