1. **State the significant challenges in visualizing big data and how to overcome these challenges. SVQCH MSFUD**

Visualizing big data is a challenging task due to the sheer volume, variety, and complexity of data involved. Some of the significant challenges in visualizing big data are:

Scalability
Data variety.
Data quality
Data complexity
High performance requirements

To overcome these challenges, the following steps can be taken:

Meeting the need for speed
Simplify the Data
Filter and Subset Data
Use Appropriate Visualization Techniques
Data Cleaning and Quality Improvement

2. **Identify the technique used to evaluate the performance of a model on unseen data. Also list all its types. (Evaluation performance cross validation technique)**

Cross-Validation is a resampling technique with the fundamental idea of splitting the dataset into 2 parts- training data and test data. Train data is used to train the model and the unseen test data is used for prediction. If the model performs well over the test data and gives good accuracy, it means the model hasn't overfitted the training data and can be used for prediction.

<u>Validation Set Approach</u>
We divide our input dataset into a training set and test or validation set in the validation set approach. Both the subsets are given 50% of the dataset

<u>Holdout Method</u>
This method is the simplest cross-validation technique among all. In this method, we need to remove a subset of the training data and use it to get prediction results by training it on the rest of the dataset.

<u>Leave-P-out cross-validation</u>
In this approach, the p datasets are left out of the training data. It means, if there are total n data points in the original input dataset, then n-p data points will be used as the training dataset and the p data points as the validation set. This complete process is repeated for all the samples, and the average error is calculated to know the effectiveness of the model.

<u>Leave one out cross-validation</u>
This method is similar to the leave-p-out cross-validation, but instead of p, we need to take 1 dataset out of training. It means, in this approach, for each learning set, only one datapoint is reserved, and the remaining dataset is used to train the model. This process repeats for

each datapoint.

K-Fold Cross-Validation
K-fold cross-validation approach divides the input dataset into K groups of samples of equal sizes. These samples are called **folds**. For each learning set, the prediction function uses k-1 folds, and the rest of the folds are used for the test set.

Stratified k-fold cross-validation
This technique is similar to k-fold cross-validation with some little changes. This approach works on stratification concept, it is a process of rearranging the data to ensure that each fold or group is a good representative of the complete dataset. To deal with the bias and variance, it is one of the best approaches.

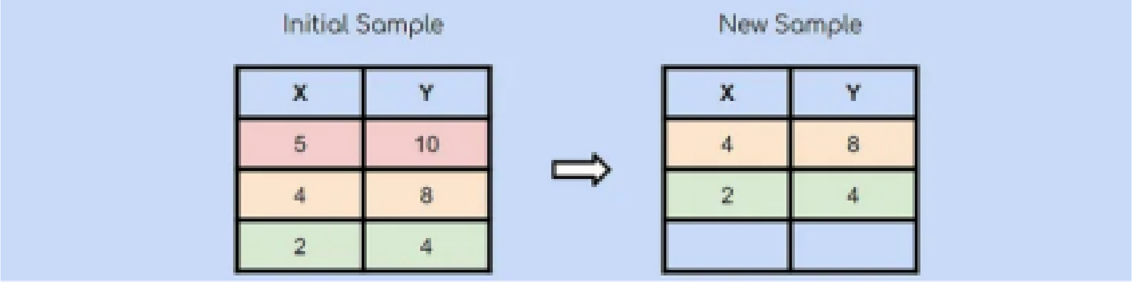
Time-series cross-validation: This method is used for time-series data, where the data is split into training and testing sets in a sequential manner, with the testing set always following the training set.

Repeated cross-validation: In this method, the k-fold cross-validation process is repeated multiple times, with different random splits of the data, to obtain a more robust estimate of model performance.

3. Discuss the technique used to perform random sampling with replacement . State its advantages and disadvantages (Bootstrapping)

Bootstrapping is a statistical technique used for random sampling with replacement. It involves taking random samples from a dataset and using those samples to estimate the variability of a statistic of interest, such as the mean or standard deviation. The samples are taken with replacement, meaning that each observation has an equal chance of being selected for each sample.

Here's how bootstrapping works:
1. Starting with the original dataset, a large number of "bootstrap samples" are created by randomly sampling observations from the original dataset with replacement. 2. For each bootstrap sample, the statistical estimator or model is calculated and recorded.
3. The distribution of the estimator or model across all of the bootstrap samples is used to estimate the uncertainty or variability of the estimator or model.

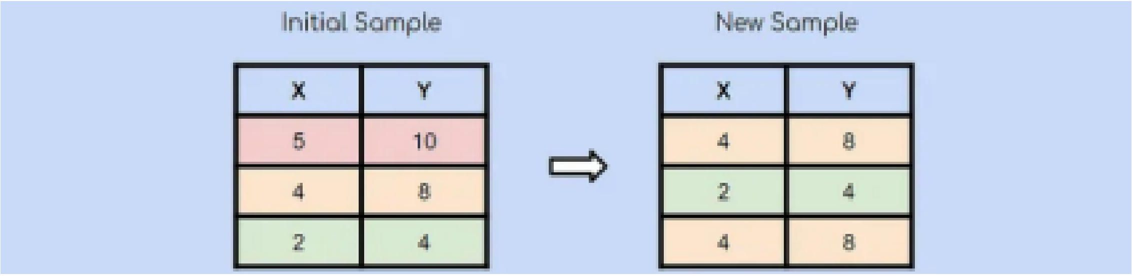


Initial Sample

X	Y
5	10
4	8
2	4

New Sample

X	Y
4	8
2	4



Initial Sample

X	Y
5	10
4	8
2	4

New Sample

X	Y
4	8
2	4
4	8

Advantages of bootstrapping: NSFAR

- Non-parametric.
- Simplicity
- Flexibility
- Accuracy
- Robustness

Disadvantages of bootstrapping: CBSD

- Computationally intensive
- Bias
- Sampling variability
- Dependence on random number generator

4. State the causes of the outlier. MNDSU

Measurement error: Outliers can be caused by errors in data collection or measurement. For example, a sensor malfunction or human error can lead to an extreme value being recorded.

Natural variation: In some cases, outliers may occur due to natural variation in the data. For example, in a distribution with a long tail, there may be some extreme values that are not necessarily errors but reflect the true variability of the data.

Data processing errors: Outliers can also be caused by errors in data processing, such as data entry or data manipulation.

Sampling error: Outliers may also occur due to sampling error, which is the result of selecting a sample that is not representative of the population.

Unusual events: Outliers may be caused by unusual or rare events that are not representative of the typical data. For example, a stock market crash or a natural disaster can lead to extreme values in financial or environmental data.

5. State the Impact of the outlier. SIIRBD

Skewness
Influence on model parameters
Increased variability

Reduced power
Bias
Decreased model performance
.

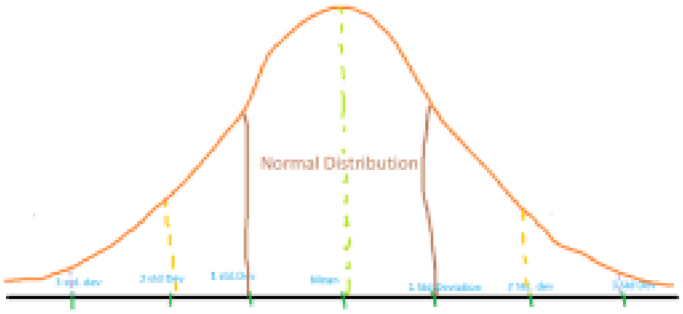
6. Demonstrate the Methods to identify outliers with an example.

Statistical methods:

Statistical methods use various measures of *central tendency* and *dispersion*, such as mean, median, standard deviation, quartiles, and interquartile range, to identify outliers. The most commonly used statistical method is the Z-score method, which uses the standard deviation to identify data points that fall outside a specified range.

Z score is an important concept in statistics. Z score is also called standard score. This score helps to understand if a data value is greater or smaller than mean and how far away it is from the mean. More specifically, Z score tells how many standard deviations away a data point is from the mean.

A normal distribution is shown below and it is estimated that
68% of the data points lie between +/- 1 standard deviation.
95% of the data points lie between +/- 2 standard deviation
99.7% of the data points lie between +/- 3 standard deviation



If the z score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be an outlier. For example, in a survey, it was asked how many children a person had. Suppose the data obtained from people is

1, 2, 2, 2, 3, 1, 1, 15, 2, 2, 2, 3, 1, 1, 2

mean of the dataset is 2.67
std. deviation is 3.36

z = (i-mean)/std = 15-2.67/3.35 = 3.7 > 3

Therefore, 15 is an outlier

Distance-based methods:

Distance-based methods use distance measures such as Euclidean distance, Mahalanobis distance, and Manhattan distance to identify outliers. The most commonly used distance-based method is the k-nearest neighbor (k-NN) method, which identifies outliers as data points that are farthest away from their k-nearest neighbors.

Although kNN is a supervised ML algorithm, when it comes to anomaly detection it takes an unsupervised approach. This is because there is no actual “learning” involved in the process and there is no predetermined labeling of “outlier” or “not-outlier” in the dataset, instead, it is entirely based upon threshold values.

Suppose we have a dataset of 5 points in two dimensions:

(2, 3), (4, 5), (6, 7), (8, 9), and (10, 11)

We want to detect if any points are outliers based on their Euclidean distance from the mean of the dataset. The mean is calculated as follows:

mean_x = (2 + 4 + 6 + 8 + 10) / 5 = 6
mean_y = (3 + 5 + 7 + 9 + 11) / 5 = 7

So, the mean of the dataset is (6, 7).

Next, we can calculate the Euclidean distance of each point from the mean:

Euclidean distance of (2, 3) from (6, 7) = sqrt((6 - 2)^2 + (7 - 3)^2) = 5.66
Euclidean distance of (4, 5) from (6, 7) = sqrt((6 - 4)^2 + (7 - 5)^2) = 2.83
Euclidean distance of (6, 7) from (6, 7) = 0
Euclidean distance of (8, 9) from (6, 7) = sqrt((6 - 8)^2 + (7 - 9)^2) = 2.83
Euclidean distance of (10, 11) from (6, 7) = sqrt((6 - 10)^2 + (7 - 11)^2) = 5.66
From the above calculation, we can see that the points (2, 3) and (10, 11) are the outliers as they have the largest Euclidean distance from the mean. Therefore, we can consider them as outliers in this dataset.

Density-based methods:
Density-based methods use measures of local density to identify outliers. The most commonly used density-based method is the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, which identifies outliers as data points that are not part of any cluster or are part of a small, sparse cluster.

Suppose we have 5 data points in two-dimensional space: (1,1), (1,2), (2,1), (2,2), and (10,10). We want to detect the outlier in this data set using DBSCAN.

Here are the steps:

- Choose the parameters: We need to choose the parameters for DBSCAN. Let's choose epsilon (eps) as 1.5 and minimum points (min_samples) as 2. - Calculate distances: Calculate the distances between each data point and all other data points.
- Define core points, border points, and noise points: A point is a core point if it has at least min_samples points within a distance of eps. A point is a border point if it is not a core point but is within a distance of eps from a core point. A point is a noise point if it is neither a core point nor a border point.
- Assign labels: Assign labels to the points. Start with an arbitrary point, and if it is

a core point, assign it a new label. Then, recursively expand the cluster by adding border points to the cluster. Finally, mark noise points as outliers.

Result: In this example, points (1,1), (1,2), (2,1), and (2,2) are core points because they all have at least 2 other points within a distance of 1.5. The point (10,10) is a noise point because it does not have any other point within a distance of 1.5. Therefore, (10,10) is the outlier in this data set.

7. State the Need for anomaly detection. Specify the basic approaches to anomaly detection. Enlist the application of anomaly detection.

Anomaly detection is an important task in data analysis and machine learning. It involves identifying patterns in data that are unusual or unexpected, and can be used for a variety of applications including fraud detection, intrusion detection, fault detection, and quality control.

The need for anomaly detection arises because anomalies or outliers can have a significant impact on *statistical analyses and modeling*, and may indicate the presence of *unusual or abnormal behavior* that needs to be investigated. By identifying and removing or flagging anomalies, data analysts and machine learning engineers can improve the accuracy of their models and gain insights into potential issues.

There are several approaches to anomaly detection, including:

- Statistical Methods
- Machine Learning Methods
- Rule Based Methods
- Deep Learning Methods
- Density Based Methods
- Distance Based Methods

Some common applications of anomaly detection include: FNQPM

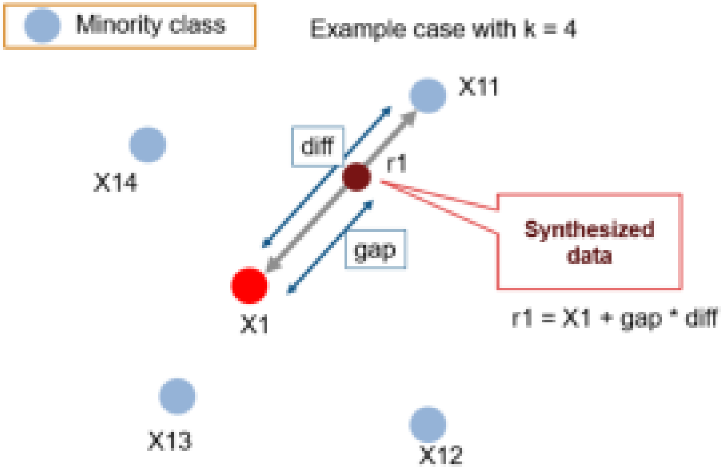
- Fraud detection
- Network intrusion detection
- Quality control
- Predictive maintenance
- Medical diagnosis

8. Discuss the algorithm which will overcome the overfitting problem posed by random oversampling with an example. (SMOT)

SMOTE (Synthetic Minority Over-sampling Technique) is an algorithm used for oversampling in imbalanced datasets. It addresses the overfitting problem posed by random oversampling by generating *synthetic samples rather than duplicating existing* ones. SMOTE works by selecting a minority class sample and finding its *k nearest minority class neighbors*. It focuses on the feature space to generate new instances with the help of *interpolation* between the positive instances that lie together. It then generates new minority class samples along the line segments joining these neighbors.

For example, consider a dataset with two classes: Class A (minority) and Class B (majority). The dataset has 100 samples, with 10 samples belonging to Class A and 90 samples belonging to Class B. This is an imbalanced dataset, with Class A being the minority class.

Using SMOTE, new minority class samples are generated by selecting a minority class sample and finding its k nearest minority class neighbors. Suppose k=4, and the algorithm selects sample X1 as the starting point. The 4 nearest minority class neighbors are X11, X12, X13, X14. SMOTE generates new samples along the line segments joining X1 to each of its neighbors, resulting in 4 new synthetic minority class samples.



The advantage of SMOTE over random oversampling is that it *reduces the risk of overfitting* by generating synthetic samples that are not exact copies of existing samples. This can *improve the generalization* of machine learning models trained on imbalanced datasets.

However, SMOTE may also *introduce some noise* into the dataset, especially if the k nearest neighbors are not well-chosen. In addition, SMOTE may not work well for datasets with high dimensionality or non-linear decision boundaries.

To overcome these limitations, variants of SMOTE have been proposed, such as *Borderline SMOTE*, which only generates synthetic samples for borderline instances, and *ADASYN*, which adaptively generates synthetic samples *based on the density* of minority class instances.

Module 5 questions: Time series forecasting

1) **What are the methods of time series decomposition? When to use one method over other**

Time series decomposition involves thinking of a series as a combination of level, trend, seasonality, and noise components.
Decomposition provides a useful abstract model for thinking about time series generally and for better understanding problems during time series analysis and forecasting. A useful abstraction for selecting forecasting methods is to break a time series down into systematic and unsystematic components.

Systematic: Components of the time series that have consistency or recurrence and can be described and modeled.
Non-Systematic: Components of the time series that cannot be directly modeled. A given time series is thought to consist of three systematic components including level, trend, seasonality, and one non-systematic component called noise.

These components are defined as follows:
Level: The average value in the series.
Trend: The increasing or decreasing value in the series.
Seasonality: The repeating short-term cycle in the series.
Noise: The random variation in the series.

A series is thought to be an aggregate or combination of these four components. All series have a level and noise. The trend and seasonality components are optional. It is helpful to think of the components as combining either additively or multiplicatively.

Additive Decomposition
An additive model suggests that the components are added together as follows: y(t) = Level + Trend + Seasonality + Noise
An additive model is linear where changes over time are consistently made by the same amount.
A linear trend is a straight line.
A linear seasonality has the same frequency (width of cycles) and amplitude (height of cycles).

Multiplicative Decomposition
A multiplicative model suggests that the components are multiplied together as follows:
y(t) = Level * Trend * Seasonality * Noise
A multiplicative model is nonlinear, such as quadratic or exponential. Changes increase or decrease over time.
A nonlinear trend is a curved line.
A non-linear seasonality has an increasing or decreasing frequency and/or amplitude over time.

Pseudo Additive Decomposition
Pseudo Additive models combine elements of additive and multiplicative models. Useful when time series value are close to or equal to zero and you require a multiplicative model
y(t) = Trend + Trend(Seasonality - 1) + Trend(Noise -1) = Trend(Seasonality + Noise - 1)

Additive decomposition is typically used when the magnitude of seasonality does not depend on the level of the time series.
Multiplicative decomposition is typically used when the magnitude of seasonality increases with the level of the time series
If the magnitude of seasonality remains constant over time, additive decomposition may be more appropriate. On the other hand, if the magnitude of seasonality increases or decreases with the level of the time series, multiplicative decomposition may be more appropriate.

2) Describe in detail ARIMA model

ARIMA (AutoRegressive Integrated Moving Average) model is a popular time series forecasting method that combines the autoregressive (AR) and moving average (MA) models with differencing to account for non-stationarity in the time series data. ARIMA models are widely used in various fields such as economics, finance, and engineering to forecast future values of a time series based on its historical data.

The ARIMA model consists of three components:
Autoregressive (AR) component: This component models the linear relationship between the current observation and the previous p observations (p is the order of the autoregressive component). The AR component uses a regression equation that includes the past values of the time series, weighted by their respective coefficients.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

where,
yt is the time series observation at time t
c is a constant
ϕ1, ϕ2, … , ϕp are the autoregressive coefficients
εt is the white noise error term at time t

Moving Average (MA) component: This component models the linear relationship between the current observation and the previous q forecast errors (q is the order of the moving average component). The MA component uses a regression equation that includes the past forecast errors, weighted by their respective coefficients.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

where,
θ1, θ2, … , θq are the moving average coefficients

Integrated (I) component: This component models the differencing required to make the time series stationary. Stationarity is a desirable property of a time series that ensures that its statistical properties, such as mean and variance, are constant over time. The I component involves taking the difference between the time series and its lagged value until it becomes stationary.

$$yt = (1-B)^d Z_t$$

where,
B is the backshift operator
Zt is the differenced time series
d is the degree of differencing

The ARIMA model is denoted as ARIMA(p,d,q),
p: the number of lag observations in the model, also known as the lag order. d: the

number of times the raw observations are differenced; also known as the degree of differencing.

q: the size of the moving average window, also known as the order of the moving average.

The parameters of the ARIMA model are estimated using the *maximum likelihood estimation* method based on the historical data. Once the parameters are estimated, the ARIMA model can be used to forecast future values of the time series.

The ARIMA model is fitted by estimating the values of the model parameters (ϕ1, ϕ2, … , ϕp, θ1, θ2, … , θq, d) that minimize the sum of squared errors between the actual and predicted values. The estimated model can then be used to forecast future values of the time series. The ARIMA model has several advantages, including its ability to handle both stationary and non-stationary time series, its flexibility to incorporate external factors, and its ability to provide probabilistic forecasts. However, the ARIMA model also has some limitations, such as its assumption of linear relationships between variables and its sensitivity to outliers.

3) Previous data of the rainfall from year 2000 to 2020 is available with us, we need to predict monthly rainfall for future years. State and explain the ideal model used for this case study.

For predicting monthly rainfall for future years, the ideal model to use is a seasonal ARIMA (SARIMA) model.

The SARIMA model is an extension of the ARIMA model that includes seasonal components to account for periodic patterns in the time series, such as monthly or quarterly seasonality. In the case of rainfall data, there is likely to be a seasonal pattern due to the annual cycle of weather patterns.

The SARIMA model is denoted as SARIMA(p,d,q)(P,D,Q)$_m$, where p, d, and q are the orders of the non-seasonal AR, I, and MA components, P, D, and Q are the orders of the seasonal AR, I, and MA components, and m is the number of periods in a season (i.e., 12 for monthly data). The seasonal components are similar to the non-seasonal components, but they operate over a seasonal lag instead of a non-seasonal lag.

The seasonal AR component equation is given by:

$$y_t = c + \phi_1 y_{t-m} + \phi_2 y_{t-2m} + \dots + \phi_P y_{t-Pm} + \varepsilon_t$$

where,

P is the number of seasonal lags

ϕ1, ϕ2, … , ϕP are the seasonal autoregressive coefficients

The seasonal MA component equation is given by:

$$y_t = c + \varepsilon_t + \phi_1 \varepsilon_{t-m} + \phi_2 \varepsilon_{t-2m} + \dots + \phi_Q \varepsilon_{t-Qm}$$

where,

Q is the number of seasonal moving averages

ϕ1, ϕ2, … , ϕQ are the seasonal moving average coefficients

The seasonal I component equation is given by:

$$y_t = (1 - B^m)^D (Z_t - Z_{t-m})$$

where,

B is the backshift operator
Zt is the differenced time series
D is the degree of seasonal differencing

The SARIMA model is fitted by estimating the values of the model parameters (ϕ1, ϕ2, … , ϕp, θ1, θ2, … , θq, d, P, D, Q) that minimize the sum of squared errors between the actual and predicted values. The estimated model can then be used to forecast future values of the time series.

To build a SARIMA model for the rainfall data, we would first need to identify the optimal values of p, d, q, P, D, Q, and m through a process called model selection. This involves analyzing the *autocorrelation* and *partial autocorrelation* functions of the time series data and selecting the model with the lowest *Akaike Information Criterion* (AIC) or *Bayesian Information Criterion* (BIC) value.

Once the optimal SARIMA model is selected, we can use it to make forecasts of future monthly rainfall. The SARIMA model will take into account both the non-seasonal and seasonal components of the time series and can provide accurate forecasts for different time horizons.

Overall, the SARIMA model is an ideal model for predicting monthly rainfall for future years because it can capture the seasonal patterns in the data and account for the non-stationarity of the time series.

4) **Illustrate various smoothing methods applied on the time series data with an example.**

Naive, Seasonal naive, Average, moving average, weighted average, exponential average
Naive Method: This is a simple method that involves using the most recent observation as the forecast for the next period. For example, if we have a time series of monthly sales data, the forecast for next month would be the sales value for the current month.

Seasonal Naive Method: Similar to the naive method, but instead of using the most recent observation, it uses the corresponding observation from the previous season. For example, if we have a time series of quarterly sales data, the forecast for next quarter would be the sales value for the same quarter in the previous year.

Simple Average Method: This method involves taking the average of all the past observations and using it as the forecast for the next period. For example, if we have a time series of daily temperature data, we could calculate the average temperature for the past 30 days and use it as the forecast for tomorrow's temperature.

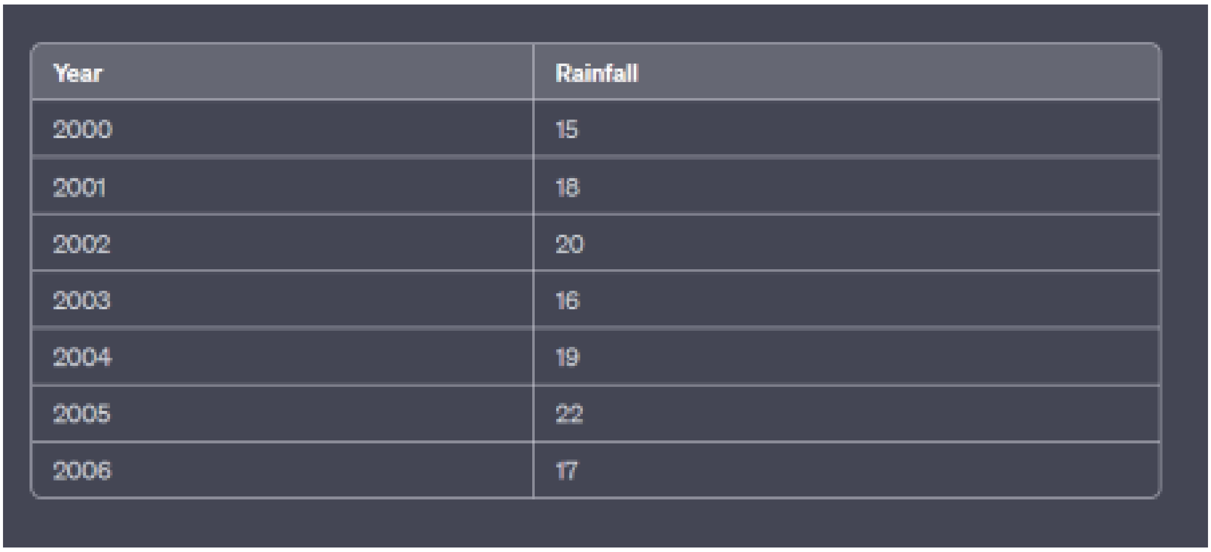
Moving Average Method: Similar to the simple average method, but only considers a window of the most recent observations. For example, if we have a time series of hourly website traffic data, we could calculate a 7-day moving average by taking the average of the traffic values for the current hour and the previous 167 hours.

Weighted Moving Average Method: Similar to the moving average method, but assigns different weights to each observation in the window. The weights can be used to emphasize or de-emphasize certain observations based on their relative importance.

Exponential Smoothing Method: This method involves assigning exponentially decreasing weights to past observations, with the most recent observations given the highest weights. The weights are determined by a smoothing factor, which controls how quickly the weights decrease. This method is particularly useful for data with trend and seasonality.

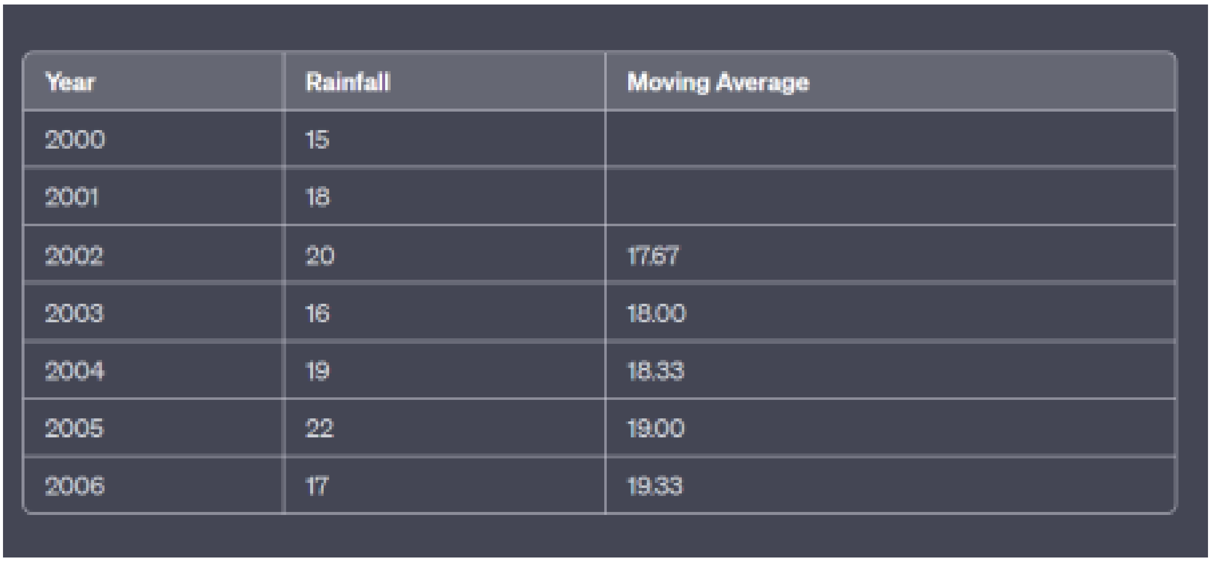
Smoothing methods are commonly used in time series analysis to remove noise and identify underlying trends or patterns in the data. Some of the common smoothing methods used in time series analysis are:

Moving Average (MA) Smoothing: This method involves calculating the average of a fixed number of consecutive data points and using this average value as the smoothed value. The size of the moving window or the number of data points to be averaged is determined by the analyst. For example, a 3-period moving average can be calculated for the following time series data:



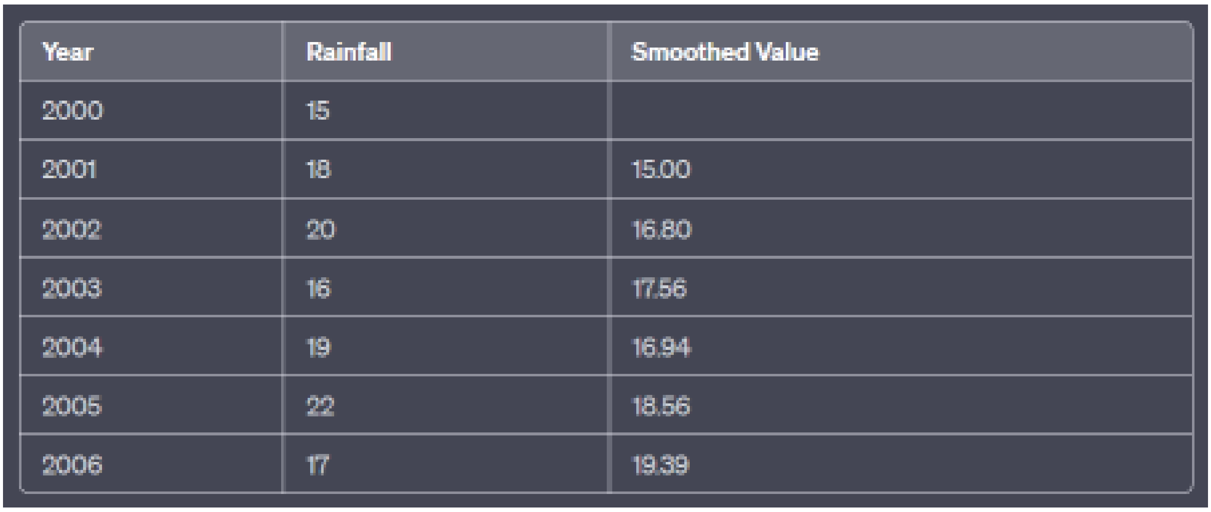
Year	Rainfall
2000	15
2001	18
2002	20
2003	16
2004	19
2005	22
2006	17

The 3-period moving average for this time series can be calculated as follows:



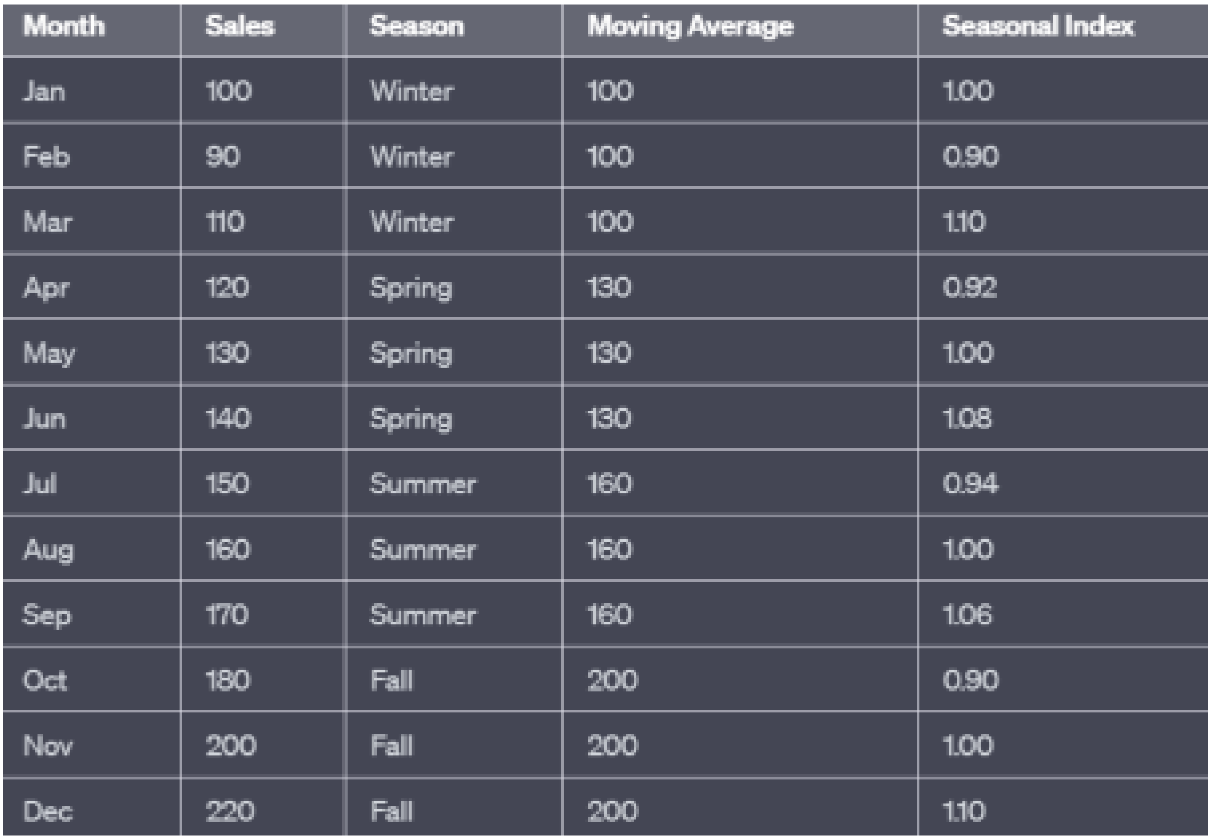
Year	Rainfall	Moving Average
2000	15	
2001	18	
2002	20	17.67
2003	16	18.00
2004	19	18.33
2005	22	19.00
2006	17	19.33

Exponential Smoothing: This method involves calculating a weighted average of past observations, with more weight given to recent observations. The weights are determined by a smoothing parameter, which is usually between 0 and 1. For example, an exponential smoothing model can be applied to the same rainfall time series data with a smoothing parameter of 0.3, as follows:

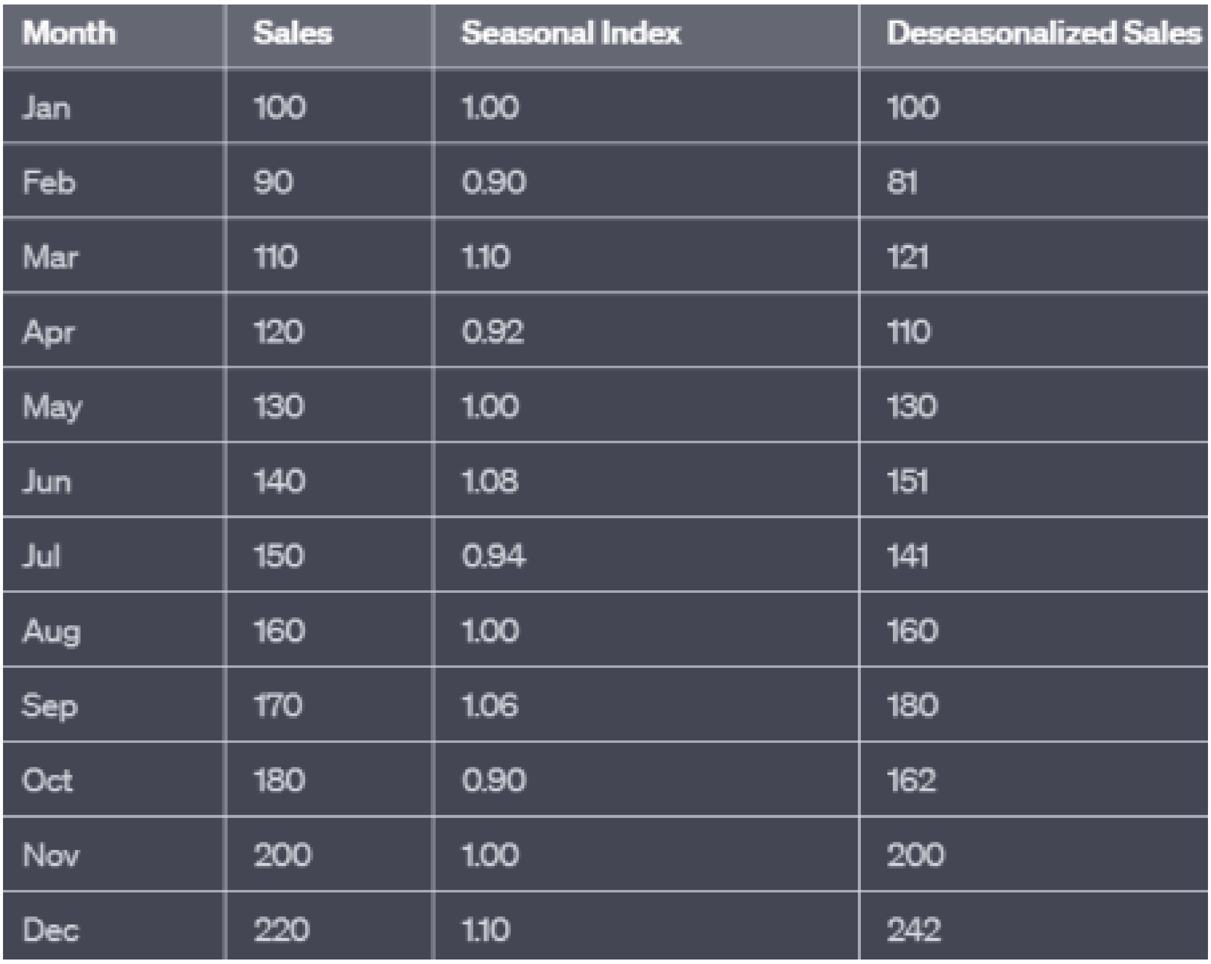


Year	Rainfall	Smoothed Value
2000	15	
2001	18	15.00
2002	20	16.80
2003	16	17.56
2004	19	16.94
2005	22	18.56
2006	17	19.39

Seasonal Smoothing: This method is used to remove seasonality from the time series data. Seasonal smoothing involves taking a moving average of the data within each season. For example, a seasonal smoothing model can be applied to monthly sales data for a retail store to remove the seasonal effect of Christmas sales. The smoothed value for each month would be the average of the same month's sales data over the past few years.



Month	Sales	Season	Moving Average	Seasonal Index
Jan	100	Winter	100	1.00
Feb	90	Winter	100	0.90
Mar	110	Winter	100	1.10
Apr	120	Spring	130	0.92
May	130	Spring	130	1.00
Jun	140	Spring	130	1.08
Jul	150	Summer	160	0.94
Aug	160	Summer	160	1.00
Sep	170	Summer	160	1.06
Oct	180	Fall	200	0.90
Nov	200	Fall	200	1.00
Dec	220	Fall	200	1.10



Month	Sales	Seasonal Index	Deseasonalized Sales
Jan	100	1.00	100
Feb	90	0.90	81
Mar	110	1.10	121
Apr	120	0.92	110
May	130	1.00	130
Jun	140	1.08	151
Jul	150	0.94	141
Aug	160	1.00	160
Sep	170	1.06	180
Oct	180	0.90	162
Nov	200	1.00	200
Dec	220	1.10	242

Overall, smoothing methods can help to identify underlying trends and patterns in time series data by removing noise and seasonality. The choice of a particular smoothing method depends on the nature of the time series data and the specific research question at hand.