CS5805 : Machine Learning I
Lecture #3

Reza Jafari, Ph.D

Collegiate Associate Professor
rjafari@vt.edu

COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH.

What is a Data Mining?

- Data mining provides a way for a computer to learn how to make decisions with data and forms the backbone of many high-tech systems of today.

What is a Data Mining?

- Data mining provides a way for a computer to learn how to make decisions with data and forms the backbone of many high-tech systems of today.
- This decision could be predicting tomorrow's weather, blocking a spam email, detecting the language of a web site, finding a new romance on a dating site,...

What is a Data Mining?

- Data mining provides a way for a computer to learn how to make decisions with data and forms the backbone of many high-tech systems of today.
- This decision could be predicting tomorrow's weather, blocking a spam email, detecting the language of a web site, finding a new romance on a dating site,...
- Data mining is part of statistics, engineering, optimization, and computer science.

What is a Data Mining?

- Data mining provides a way for a computer to learn how to make decisions with data and forms the backbone of many high-tech systems of today.
- This decision could be predicting tomorrow's weather, blocking a spam email, detecting the language of a web site, finding a new romance on a dating site,...
- Data mining is part of statistics, engineering, optimization, and computer science.
- We start our data mining process by creating a Dataset, describing an aspect of the real world.

Data objects

record, *point*, *vector*, *pattern*, *event*, *case*, *sample*, *instance*, *observation* or *entity*.

Attributes

variable, *field*, *feature* or *dimension*.

What is a Dataset?

- Collection of Data Objects and Attributes.

What is a Dataset?

- Collection of Data Objects and Attributes.
- An Attribute is a property or characteristic of an object:

Attributes

Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

What is a Dataset?

- Collection of Data Objects and Attributes.
- An Attribute is a property or characteristic of an object:
  - Eye color of a person, temperature, Tax income, Marital Status,..

Attributes

Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

What is a Dataset?

- Collection of Data Objects and Attributes.
- An Attribute is a property or characteristic of an object:
  - Eye color of a person, temperature, Tax income, Marital Status,..
  - A collection of attributes describe an Object.

Attributes

Objects

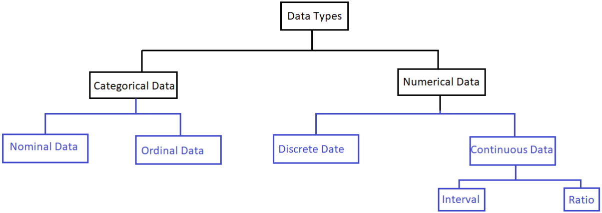
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Type of Attributes

Categorical

- Categorical data is a type of data than can be stored into groups or categories with the aid of names or labels. It is also known as qualitative data.

Numerical



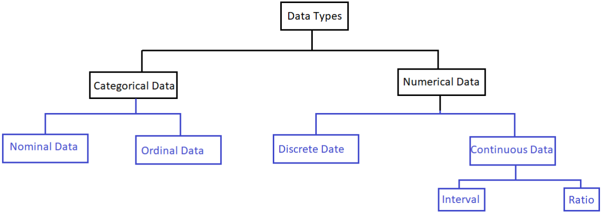
Type of Attributes

Categorical

- Categorical data is a type of data than can be stored into groups or categories with the aid of names or labels. It is also known as qualitative data.

Numerical

- Numerical data is expressed in terms of numbers and it can be collected in number form. It is also known as quantitative data.



Categorical Data

- Categorical data are mainly divided into :

Nominal

Ordinal

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.

Ordinal

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.
- Sometimes called naming data.

Ordinal

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.
- Sometimes called naming data.
- Can be counted but not measured.
- E.g. Employee ID, zip code, Name of person, gender, eye color,...

Ordinal

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.
- Sometimes called naming data.
- Can be counted but not measured.
- E.g. Employee ID, zip code, Name of person, gender, eye color,...
- Operations: mode, entropy, contingency correlation, χ^2 test.

Ordinal

- Include elements that are ranked, ordered, or rating scale.

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.
- Sometimes called naming data.
- Can be counted but not measured.
- E.g. Employee ID, zip code, Name of person, gender, eye color,...
- Operations: mode, entropy, contingency correlation, χ^2 test.

Ordinal

- Include elements that are ranked, ordered, or rating scale.
- Grades, size{small, medium, large }, hardness of materials {good, better, best}

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.
- Sometimes called naming data.
- Can be counted but not measured.
- E.g. Employee ID, zip code, Name of person, gender, eye color,...
- Operations: mode, entropy, contingency correlation, χ^2 test.

Ordinal

- Include elements that are ranked, ordered, or rating scale.
- Grades, size{small, medium, large }, hardness of materials {good, better, best}
- Rating scale of 1 to 5 for the restaurant {stronlgy dislike, dislike, neural, like, strongly like}. A rating of 5 indicates more enjoyment than a rating of 4.

Categorical Data

- Categorical data are mainly divided into :

Nominal

- Categorical data that names or labels.
- Sometimes called naming data.
- Can be counted but not measured.
- E.g. Employee ID, zip code, Name of person, gender, eye color,...
- Operations: mode, entropy, contingency correlation, χ^2 test.

Ordinal

- Include elements that are ranked, ordered, or rating scale.
- Grades, size{small, medium, large }, hardness of materials {good, better, best}
- Rating scale of 1 to 5 for the restaurant {stronlgy dislike, dislike, neural, like, strongly like}. A rating of 5 indicates more enjoyment than a rating of 4.

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

Continuous

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

- Discrete data is a type of numerical data with countable elements.

Continuous

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

- Discrete data is a type of numerical data with countable elements.
- Discrete data could be either countably finite or countably infinite. E.g, the bags of rice in store is countably finite. The grains of rice in a bag is countably infinite.
- E.g, age, number of students in class, zip code, ID number,...

Continuous

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

- Discrete data is a type of numerical data with countable elements.
- Discrete data could be either countably finite or countably infinite. E.g, the bags of rice in store is countably finite. The grains of rice in a bag is countably infinite.
- E.g, age, number of students in class, zip code, ID number,...

Continuous

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

- Discrete data is a type of numerical data with countable elements.
- Discrete data could be either countably finite or countably infinite. E.g, the bags of rice in store is countably finite. The grains of rice in a bag is countably infinite.
- E.g, age, number of students in class, zip code, ID number,...

Continuous

- Continuous data is a numerical data type with uncountable elements.

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

- Discrete data is a type of numerical data with countable elements.
- Discrete data could be either countably finite or countably infinite. E.g, the bags of rice in store is countably finite. The grains of rice in a bag is countably infinite.
- E.g, age, number of students in class, zip code, ID number,...

Continuous

- Continuous data is a numerical data type with uncountable elements.
- A set of intervals on a real number line.

Numerical Data

- Numerical data can be used as a form of measurement, such as a person's height, weight, IQ, etc. There are two types of numerical data

Discrete

- Discrete data is a type of numerical data with countable elements.
- Discrete data could be either countably finite or countably infinite. E.g, the bags of rice in store is countably finite. The grains of rice in a bag is countably infinite.
- E.g, age, number of students in class, zip code, ID number,...

Continuous

- Continuous data is a numerical data type with uncountable elements.
- A set of intervals on a real number line.
- E.g, GPA, height, weight,temperature,...

Continuous Data

- Continuous data can be further divided into interval and ratio.

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

Ratio

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.

Ratio

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.

Ratio

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.
- E.g, Calendar dates

Ratio

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.
- E.g, Calendar dates
- Operations: **arithmetic mean**, standard deviation, Pearson's correlation, t and F-tests.

Ratio

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.
- E.g, Calendar dates
- Operations: **arithmetic mean**, standard deviation, Pearson's correlation, t and F-tests.

Ratio

- Ratio has absolute zero and never fall below zero.

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in <u>economic</u> and <u>business</u>.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.
- E.g, Calendar dates
- Operations: **arithmetic mean**, standard deviation, Pearson's correlation, t and F-tests.

Ratio

- Ratio has absolute zero and never fall below zero.
- E.g, height, weight (measure from 0 and above, but never fall below it), f-score, false positive,

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.
- E.g, Calendar dates
- Operations: **arithmetic mean**, standard deviation, Pearson's correlation, t and F-tests.

Ratio

- Ratio has absolute zero and never fall below zero.
- E.g, height, weight (measure from 0 and above, but never fall below it), f-score, false positive,
- E.g, the Kelvin scale is a ratio, since 0 represents a total lack of thermal energy.

Continuous Data

- Continuous data can be further divided into interval and ratio.
- Interval and ratio data are **king** in economic and business.

Interval

- Interval scales hold NO true zero and can represent values **below zero**.
- E.g, Temperature in F or C can be measured below 0 degree, -10C.
- E.g, Calendar dates
- Operations: **arithmetic mean**, standard deviation, Pearson's correlation, t and F-tests.

Ratio

- Ratio has absolute zero and never fall below zero.
- E.g, height, weight (measure from 0 and above, but never fall below it), f-score, false positive,
- E.g, the Kelvin scale is a ratio, since 0 represents a total lack of thermal energy.

What is the Average?

- In statistics, a central tendency is a central or typical value of probability distribution.

What is the Average?

- In statistics, a central tendency is a central or typical value of probability distribution.
- In other words, it is a value that has the highest probability distribution that describes all possible values that a variable may have.

What is the Average?

- In statistics, a central tendency is a central or typical value of probability distribution.
- In other words, it is a value that has the highest probability distribution that describes all possible values that a variable may have.
- The most common measure of central tendency are the arithmetic mean, median(the middle value in the dataset) and the mode(the most frequent value in the dataset).

What is the Average?

- In statistics, a central tendency is a central or typical value of probability distribution.
- In other words, it is a value that has the highest probability distribution that describes all possible values that a variable may have.
- The most common measure of central tendency are the arithmetic mean, median(the middle value in the dataset) and the mode(the most frequent value in the dataset).
- Three common types of mean calculations are

1-Arithmetic mean

2-Geometric mean

3-Harmonic mean

Arithmetic mean

Arithmetic mean

- Arithmetic mean or simply mean is calculated as the sum of the values divided by the total number of values.

$$\boxed{\hat{\mu} = \frac{\sum_i^N x_i}{N}}$$

Arithmetic mean

Arithmetic mean

- Arithmetic mean or simply mean is calculated as the sum of the values divided by the total number of values.

$$\boxed{\hat{\mu} = \frac{\sum_i^N x_i}{N}}$$

- Appropriate when all values in the data sample have the same units of measure, e.g, all numbers are heights, or dollar, or miles, etc.

Arithmetic mean

Arithmetic mean

- Arithmetic mean or simply mean is calculated as the sum of the values divided by the total number of values.

$$\boxed{\hat{\mu} = \frac{\sum_{i}^{N} x_i}{N}}$$

- Appropriate when all values in the data sample have the same units of measure, e.g, all numbers are heights, or dollar, or miles, etc.
- Can be positive, negative, or zero.

Arithmetic mean

Arithmetic mean

- Arithmetic mean or simply mean is calculated as the sum of the values divided by the total number of values.

$$\boxed{\hat{\mu} = \frac{\sum_i^N x_i}{N}}$$

- Appropriate when all values in the data sample have the same units of measure, e.g, all numbers are heights, or dollar, or miles, etc.
- Can be positive, negative, or zero.
- Can be easily distorted if the sample of observations contains outliers.

Arithmetic mean

Arithmetic mean

- Arithmetic mean or simply mean is calculated as the sum of the values divided by the total number of values.

$$\boxed{\hat{\mu} = \frac{\sum_{i}^{N} x_i}{N}}$$

- Appropriate when all values in the data sample have the same units of measure, e.g, all numbers are heights, or dollar, or miles, etc.
- Can be positive, negative, or zero.
- Can be easily distorted if the sample of observations contains outliers.
- It can be used for Interval type of data.

Arithmetic mean

Arithmetic mean

- Arithmetic mean or simply mean is calculated as the sum of the values divided by the total number of values.

$$\boxed{\hat{\mu} = \frac{\sum_i^N x_i}{N}}$$

- Appropriate when all values in the data sample have the same units of measure, e.g, all numbers are heights, or dollar, or miles, etc.
- Can be positive, negative, or zero.
- Can be easily distorted if the sample of observations contains outliers.
- It can be used for Interval type of data.
- Python function from Numpy package : np.mean()

Geometric mean

Geometric mean

- Geometric mean is calculated as:

$$\boxed{\hat{\mu} = \sqrt[N]{\prod_{i}^{N} x_i}}$$

Geometric mean

Geometric mean

- Geometric mean is calculated as:

$$\boxed{\hat{\mu} = \sqrt[N]{\prod_{i}^{N} x_i}}$$

- Appropriate when data contains value with different units of measure, e.g. some measure are height, some are dollar and some miles.

Geometric mean

Geometric mean

- Geometric mean is calculated as:

$$\boxed{\hat{\mu} = \sqrt[N]{\prod_{i}^{N} x_i}}$$

- Appropriate when data contains value with different units of measure, e.g. some measure are height, some are dollar and some miles.
- Does not accept negative or zero values. All values must be positive.

Geometric mean

Geometric mean

- Geometric mean is calculated as:

$$\boxed{\hat{\mu} = \sqrt[N]{\prod_{i}^{N} x_i}}$$

- Appropriate when data contains value with different units of measure, e.g. some measure are height, some are dollar and some miles.
- Does not accept negative or zero values. All values must be positive.
- It can be used for Ratio type of data when absolute zero exits.

Geometric mean

Geometric mean

- Geometric mean is calculated as:

$$\boxed{\hat{\mu} = \sqrt[N]{\prod_{i}^{N} x_i}}$$

- Appropriate when data contains value with different units of measure, e.g. some measure are height, some are dollar and some miles.
- Does not accept negative or zero values. All values must be positive.
- It can be used for Ratio type of data when absolute zero exits.
- Application in machine learning, G-score - model evaluation metric.

$$G = \sqrt{sensitivity * specificity}$$

Geometric mean

Geometric mean

- Geometric mean is calculated as:

$$\boxed{\hat{\mu} = \sqrt[N]{\prod_{i}^{N} x_i}}$$

- Appropriate when data contains value with different units of measure, e.g. some measure are height, some are dollar and some miles.
- Does not accept negative or zero values. All values must be positive.
- It can be used for Ratio type of data when absolute zero exits.
- Application in machine learning, G-score - model evaluation metric.

$$G = \sqrt{sensitivity * specificity}$$

- The gmean function in Python scipy.stats.

Harmonic mean

Harmonic mean

- Harmonic mean is calculated as:

$$\boxed{\hat{\mu} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + ... + \frac{1}{x_N}}}$$

Harmonic mean

Harmonic mean

- Harmonic mean is calculated as:

$$\boxed{\hat{\mu} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + ... + \frac{1}{x_N}}}$$

- Appropriate when data is comprised of rates, e.g. speed, acceleration, frequency, etc.

Harmonic mean

Harmonic mean

- Harmonic mean is calculated as:

$$\boxed{\hat{\mu} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_N}}}$$

- Appropriate when data is comprised of rates, e.g. speed, acceleration, frequency, etc.
- E.g, in machine learning, when evaluating models such as True Positive or false positive rate in prediction.

Harmonic mean

Harmonic mean

- Harmonic mean is calculated as:

$$\boxed{\hat{\mu} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_N}}}$$

- Appropriate when data is comprised of rates, e.g. speed, acceleration, frequency, etc.
- E.g, in machine learning, when evaluating models such as True Positive or false positive rate in prediction.
- Does not accept negative or zero values. All values must be positive.

Harmonic mean

Harmonic mean

- Harmonic mean is calculated as:

$$\boxed{\hat{\mu} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + ... + \frac{1}{x_N}}}$$

- Appropriate when data is comprised of rates, e.g. speed, acceleration, frequency, etc.
- E.g, in machine learning, when evaluating models such as True Positive or false positive rate in prediction.
- Does not accept negative or zero values. All values must be positive.
- Application in machine learning, F-score - model evaluation metric.

$$F_1 = 2\frac{precision * recall}{precision + recall}$$

Harmonic mean

Harmonic mean

- Harmonic mean is calculated as:

$$\boxed{\hat{\mu} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + ... + \frac{1}{x_N}}}$$

- Appropriate when data is comprised of rates, e.g. speed, acceleration, frequency, etc.
- E.g, in machine learning, when evaluating models such as True Positive or false positive rate in prediction.
- Does not accept negative or zero values. All values must be positive.
- Application in machine learning, F-score - model evaluation metric.

$$F_1 = 2\frac{precision * recall}{precision + recall}$$

- The hmean function in Python scipy.stats.

In class Assignment

- Consider the following make-up dataset:

$$data = [1, 2, 3, 40, 50, 60, 10000]$$

In class Assignment

- Consider the following make-up dataset:

$$data = [1, 2, 3, 40, 50, 60, 10000]$$

- Calculate the Arithmetic mean, Geometric mean and Harmonic mean.

Confusion Matrix

- The performance of a model(classifier) can be evaluated by comparing the predicted labels against the true labels of instances.

Confusion Matrix

- The performance of a model(classifier) can be evaluated by comparing the predicted labels against the true labels of instances.
- This information can be summarized in a table called Confusion matrix. Let consider a binary classification:

		Predicted CLass	
		Class=1	Class=0
Actual Class	Class=1	f_{11}	f_{10}
	Class=0	f_{01}	f_{00}

Accuracy

$$Accuracy = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Error rate

$$Error\ rate = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Confusion Matrix

		Test result	
		Sick (positive)	Healthy (negative)
Actual	Sick	T.P	F.N
	Healthy	F.P	T.N

Sensitivity (Recall, Hit rate, true positive rate)

Number of correct positive predictions divided by the total number of positives.

$$TPR = \frac{T.P}{T.P + F.N}$$

Specificity (selectivity, true negative rate)

Number of correct negative predictions divided by the total number of negatives.

$$TNR = \frac{T.N}{T.N + F.P}$$

Confusion Matrix

- Precision: Out of all the positive predicted, what percentage is truly positive.

$$Precision = \frac{TP}{TP + FP}$$

which is a value between 0 and 1.

Confusion Matrix

- Precision: Out of all the positive predicted, what percentage is truly positive.

$$Precision = \frac{TP}{TP + FP}$$

  which is a value between 0 and 1.

- Recall or sensitivity: Out of the total positive, what percentage are predicted positive.

$$Recall = \frac{TP}{TP + FN}$$

Confusion Matrix

- Precision: Out of all the positive predicted, what percentage is truly positive.

$$Precision = \frac{TP}{TP + FP}$$

which is a value between 0 and 1.

- Recall or sensitivity: Out of the total positive, what percentage are predicted positive.

$$Recall = \frac{TP}{TP + FN}$$

- Sometimes we are looking for high Precision but sometimes looking for high Recall.

Confusion Matrix- Examples

Credit card fraud detection

- We don't want to miss any fraud transactions. Therefore we want False Negative to be as low as possible.

Spam Detection

Confusion Matrix- Examples

Credit card fraud detection

- We don't want to miss any fraud transactions. Therefore we want False Negative to be as low as possible.
- This means Recall should be **High**.

Spam Detection

Confusion Matrix- Examples

Credit card fraud detection

- We don't want to miss any fraud transactions. Therefore we want False Negative to be as low as possible.
- This means Recall should be **High**.
- Similarly, in the medical application, we don't want to miss any patient. Therefore we focus on having a high recall.

Spam Detection

Confusion Matrix- Examples

Credit card fraud detection

- We don't want to miss any fraud transactions. Therefore we want False Negative to be as low as possible.
- This means Recall should be **High**.
- Similarly, in the medical application, we don't want to miss any patient. Therefore we focus on having a high recall.

Spam Detection

- In the detection of spam mail, it is okay if any spam mail remains undetected (false negative), but what if we miss any critical mail because it is classified as spam (false positive).

Confusion Matrix- Examples

Credit card fraud detection

- We don't want to miss any fraud transactions. Therefore we want False Negative to be as low as possible.
- This means Recall should be **High**.
- Similarly, in the medical application, we don't want to miss any patient. Therefore we focus on having a high recall.

Spam Detection

- In the detection of spam mail, it is okay if any spam mail remains undetected (false negative), but what if we miss any critical mail because it is classified as spam (false positive).
- This means False Positive should be low and **Precision High**.

F1 Score

- F1 score is the harmonic mean of precision and recall.

F1 Score

- F1 score is the harmonic mean of precision and recall.
- It takes both false positive and false negatives into account.

F1 Score

- F1 score is the <u>harmonic</u> mean of precision and recall.
- It takes both false positive and false negatives into account.
- Therefore, it performs well on an imbalanced dataset.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

F1 Score

- F1 score is the harmonic mean of precision and recall.
- It takes both false positive and false negatives into account.
- Therefore, it performs well on an imbalanced dataset.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

- F1 score gives the same weightage to recall and precision.

Characteristics of Dataset

- Three characteristics of dataset that have significant impact on data mining techniques: Dimensionality, Distribution and Resolution.

Dimensionality

- Dimensionality of a dataset is the **number of attributes** that the objects possess in the dataset.

Characteristics of Dataset

- Three characteristics of dataset that have significant impact on data mining techniques: Dimensionality, Distribution and Resolution.

Dimensionality

- Dimensionality of a dataset is the **number of attributes** that the objects possess in the dataset.
- Each feature adds an extra dimension to our model.
- Analyzing data with a small number of dimensions tends to be qualitatively different from analyzing moderate or high-dimensions.

Characteristics of Dataset

- Three characteristics of dataset that have significant impact on data mining techniques: Dimensionality, Distribution and Resolution.

Dimensionality

- Dimensionality of a dataset is the **number of attributes** that the objects possess in the dataset.
- Each feature adds an extra dimension to our model.
- Analyzing data with a small number of dimensions tends to be qualitatively different from analyzing moderate or high-dimensions.
- Curse of dimensionality is a problem that occurs when data has **too many features**.

Characteristics of Dataset

- Three characteristics of dataset that have significant impact on data mining techniques: Dimensionality, Distribution and Resolution.

Dimensionality

- Dimensionality of a dataset is the **number of attributes** that the objects possess in the dataset.
- Each feature adds an extra dimension to our model.
- Analyzing data with a small number of dimensions tends to be qualitatively different from analyzing moderate or high-dimensions.
- Curse of dimensionality is a problem that occurs when data has **too many features**.
    - Over-fitting our model - results in poor performance.
- Because of above issue, dimensionality reduction is an important motivation in preprocessing.

Characteristics of Dataset

- Three characteristics of dataset that have significant impact on data mining techniques: Dimensionality, Distribution and Resolution.

Dimensionality

- Dimensionality of a dataset is the **number of attributes** that the objects possess in the dataset.
- Each feature adds an extra dimension to our model.
- Analyzing data with a small number of dimensions tends to be qualitatively different from analyzing moderate or high-dimensions.
- Curse of dimensionality is a problem that occurs when data has **too many features**.
  - Over-fitting our model - results in poor performance.
  - Harder to cluster.
- Because of above issue, dimensionality reduction is an important motivation in preprocessing.

Characteristics of Dataset

Distribution

- Distribution of a dataset is the frequency of occupance of various values for the attributes comprising data objects.

Resolution

- It is possible to obtain data at different resolutions.

Characteristics of Dataset

Distribution

- Distribution of a dataset is the frequency of occupance of various values for the attributes comprising data objects.
- Description of the concentration of objects in various regions of dataset, i.e Gaussian (Normal).

Resolution

- It is possible to obtain data at different resolutions.
- E.g, variations in atmospheric pressure on a scale of hours reflect the movement of storms. On a scale of months, such phenomena are not detectable.

Characteristics of Dataset

Distribution

- Distribution of a dataset is the frequency of occupance of various values for the attributes comprising data objects.
- Description of the concentration of objects in various regions of dataset, i.e Gaussian (Normal).
- Histogram plot.

Resolution

- It is possible to obtain data at different resolutions.
- E.g, variations in atmospheric pressure on a scale of hours reflect the movement of storms. On a scale of months, such phenomena are not detectable.

Types of Dataset

Record Data

- Data Matrix

Graph

Ordered Data

Types of Dataset

Record Data

- Data Matrix
- Document Data

Graph

Ordered Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

Ordered Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

- World Wide Web

Ordered Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

- World Wide Web
- Molecular Structures

Ordered Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

- World Wide Web
- Molecular Structures

Ordered Data

- Spatial Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

- World Wide Web
- Molecular Structures

Ordered Data

- Spatial Data
- Temporal Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

- World Wide Web
- Molecular Structures

Ordered Data

- Spatial Data
- Temporal Data
- Sequential Data

Types of Dataset

Record Data

- Data Matrix
- Document Data
- Transaction Data

Graph

- World Wide Web
- Molecular Structures

Ordered Data

- Spatial Data
- Temporal Data
- Sequential Data
- Generic Sequential Data

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of attributes.

	total_bill	tip	sex	smoker	day	time	size
0	16.99000	1.01000	Female	No	Sun	Dinner	2
1	10.34000	1.66000	Male	No	Sun	Dinner	3
2	21.01000	3.50000	Male	No	Sun	Dinner	3
3	23.68000	3.31000	Male	No	Sun	Dinner	2
4	24.59000	3.61000	Female	No	Sun	Dinner	4
5	25.29000	4.71000	Male	No	Sun	Dinner	4
6	8.77000	2.00000	Male	No	Sun	Dinner	2
7	26.88000	3.12000	Male	No	Sun	Dinner	4
8	15.04000	1.96000	Male	No	Sun	Dinner	2
9	14.78000	3.23000	Male	No	Sun	Dinner	2
10	10.27000	1.71000	Male	No	Sun	Dinner	2
11	35.26000	5.00000	Female	No	Sun	Dinner	4
12	15.42000	1.57000	Male	No	Sun	Dinner	2
13	18.43000	3.00000	Male	No	Sun	Dinner	4

Data Matrix

- Data objects with only numeric attributes can be represented by *mxn* matrix, where there are *m* rows, one for each object, and *n* columns one for each attribute.

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Data Matrix

- Data objects with only numeric attributes can be represented by *mxn* matrix, where there are *m* rows, one for each object, and *n* columns one for each attribute.
- The data objects can be thought of as points in a multi-dimensional space where each dimension represents a distinct attribute.

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Document Data

- Each document becomes a 'term' vector.

	team	coach	pla y	ball	score	game	wi n	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Document Data

- Each document becomes a 'term' vector.
- Each term is a component (attribute) of the vector.

	team	coach	pla y	ball	score	game	wi n	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Document Data

- Each document becomes a 'term' vector.
- Each term is a component (attribute) of the vector.
- The value of each component is the number of times the corresponding term occurs in the document.

	team	coach	pla y	ball	score	game	wi n	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Transaction Data

- A special type of record data where each record (transaction) involves a set of items.

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Transaction Data

- A special type of record data where each record (transaction) involves a set of items.
- Consider a grocery store.

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Transaction Data

- A special type of record data where each record (transaction) involves a set of items.
- Consider a grocery store.
- The set of products purchased by a customer during one shopping trip constitutes a transaction while the individual products that were purchased are the items.

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Transaction Data

- A special type of record data where each record (transaction) involves a set of items.

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Transaction Data

- A special type of record data where each record (transaction) involves a set of items.
- Consider a grocery store.

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

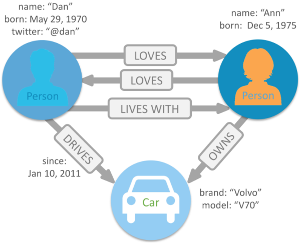
Transaction Data

- A special type of record data where each record (transaction) involves a set of items.
- Consider a grocery store.
- The set of products purchased by a customer during one shopping trip constitutes a transaction while the individual products that were purchased are the items.

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

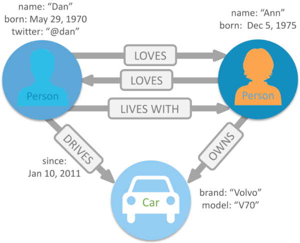
Graph Data

- We live in a connected world and understanding most domains requires processing rich sets of connections to understand what really happening.



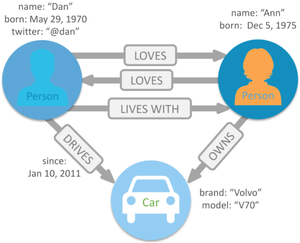
Graph Data

- We live in a connected world and understanding most domains requires processing rich sets of connections to understand what really happening.
- Connections between items are as important as the items themselves.



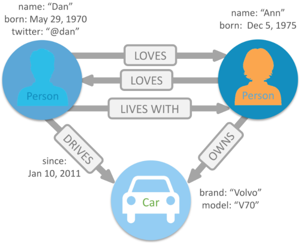
Graph Data

- We live in a connected world and understanding most domains requires processing rich sets of connections to understand what really happening.
- Connections between items are as important as the items themselves.
- <u>Nodes</u> are entities in the graph dataset. Nodes can have any number or type of relationships with sacrificing performance.



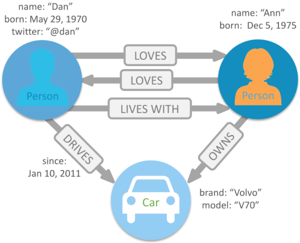
Graph Data

- We live in a connected world and understanding most domains requires processing rich sets of connections to understand what really happening.
- Connections between items are as important as the items themselves.
- Nodes are entities in the graph dataset. Nodes can have any number or type of relationships with sacrificing performance.
- Relationship provide directed, named, connections between two node entities. It has direction, start and end node.



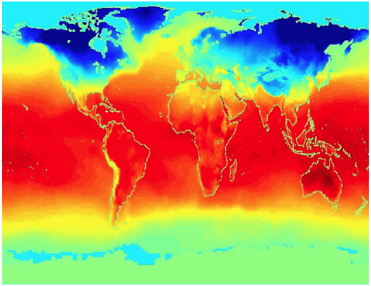
Graph Data

- We live in a connected world and understanding most domains requires processing rich sets of connections to understand what really happening.
- Connections between items are as important as the items themselves.
- Nodes are entities in the graph dataset. Nodes can have any number or type of relationships with sacrificing performance.
- Relationship provide directed, named, connections between two node entities. It has direction, start and end node.
- E.g, social network, payment networks, road networks,..



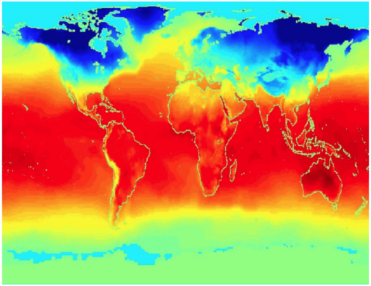
Ordered Data

- Ordered dataset have attributes with relationships that involve order in time or space.



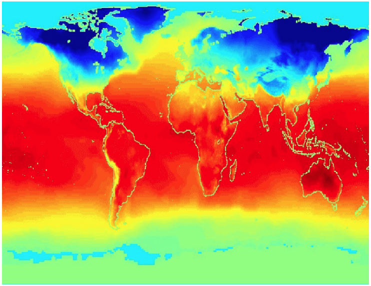
Ordered Data

- Ordered dataset have attributes with relationships that involve order in time or space.
- **Sequential transaction data**: can be thought of as an extension to transaction data, where each transaction has a time associated with.



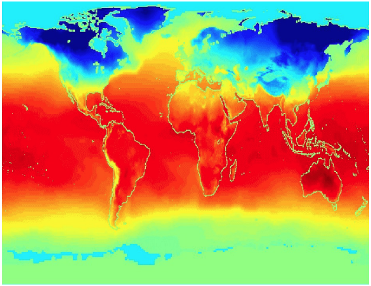
Ordered Data

- Ordered dataset have attributes with relationships that involve order in time or space.
- **Sequential transaction data**: can be thought of as an extension to transaction data, where each transaction has a time associated with.
- E.g, candy sales peak before Halloween, time is associated with each object.



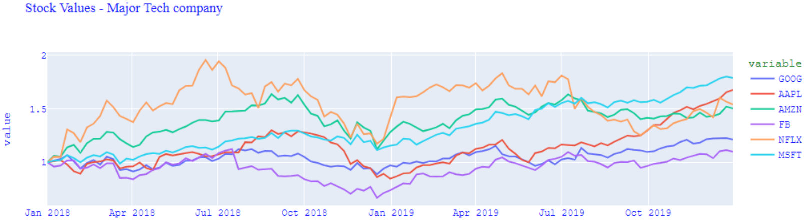
Ordered Data

- Ordered dataset have attributes with relationships that involve order in time or space.
- **Sequential transaction data**: can be thought of as an extension to transaction data, where each transaction has a time associated with.
- E.g, candy sales peak before Halloween, time is associated with each object.
- E.g, Average monthly temperature of land & ocean.



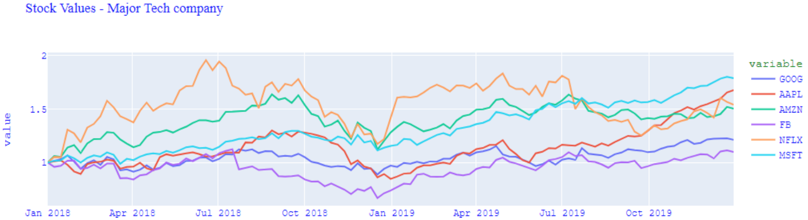
Time Series Data

- Time series data is a special type of ordered data where each record is a **time series**.



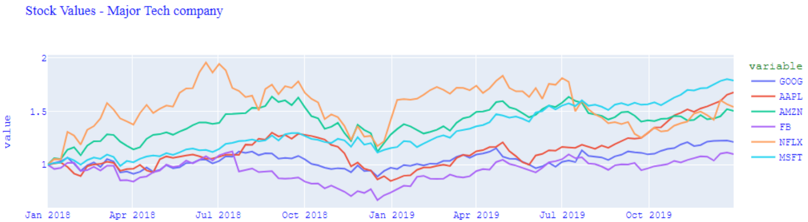
Time Series Data

- Time series data is a special type of ordered data where each record is a **time series**.
- E.g, Financial dataset might contain objects that are time series of the daily prices of stocks.



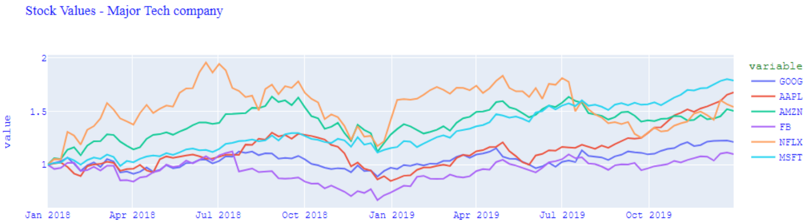
Time Series Data

- Time series data is a special type of ordered data where each record is a **time series**.
- E.g, Financial dataset might contain objects that are time series of the daily prices of stocks.
- Time domain (Autocorrelation) and Frequency domain (Power spectrum) are two contentional approaches analyzing time series dataset.



Time Series Data

- Time series data is a special type of ordered data where each record is a **time series**.
- E.g, Financial dataset might contain objects that are time series of the daily prices of stocks.
- Time domain (Autocorrelation) and Frequency domain (Power spectrum) are two contentional approaches analyzing time series dataset.
- Time series can also be modeled using Deep learning models.



Spatial and Spatio-Temporal Data

- Some objects have spatial attributes, such as positions or areas.

Spatial and Spatio-Temporal Data

- Some objects have spatial attributes, such as positions or areas.
- An example of spatial data is weather data(precipitation, temperature, pressure) that is collected for variety of geographical locations.

Spatial and Spatio-Temporal Data

- Some objects have spatial attributes, such as positions or areas.
- An example of spatial data is weather data(precipitation, temperature, pressure) that is collected for variety of geographical locations.
- Spatial-Temporal data consists of time series at various locations.

Spatial and Spatio-Temporal Data

- Some objects have spatial attributes, such as positions or areas.
- An example of spatial data is weather data(precipitation, temperature, pressure) that is collected for variety of geographical locations.
- Spatial-Temporal data consists of time series at various locations.
- An important aspects of spatial data is spatial autocorrelation, i.e. objects that are physically close tend to be similar in other way as well.

Spatial and Spatio-Temporal Data

- Some objects have spatial attributes, such as positions or areas.
- An example of spatial data is weather data(precipitation, temperature, pressure) that is collected for variety of geographical locations.
- Spatial-Temporal data consists of time series at various locations.
- An important aspects of spatial data is spatial autocorrelation, i.e. objects that are physically close tend to be similar in other way as well.
- Two points on the earth that are close o each other usually have similar temperature and rain fall.

Spatial and Spatio-Temporal Data

- Some objects have spatial attributes, such as positions or areas.
- An example of spatial data is weather data(precipitation, temperature, pressure) that is collected for variety of geographical locations.
- Spatial-Temporal data consists of time series at various locations.
- An important aspects of spatial data is spatial autocorrelation, i.e. objects that are physically close tend to be similar in other way as well.
- Two points on the earth that are close o each other usually have similar temperature and rain fall.
- An example of spatial data is weather data (precipitation, temperature, pressure) that is collected for variety of geographical locations.

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.
- There are several strategies for dealing with missing data:

Eliminate Data Objects or Attributes

- A simple and effective strategy is to eliminate objects with missing values.
- Eliminate attributes that have missing values.

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.
- There are several strategies for dealing with missing data:

Eliminate Data Objects or Attributes

- A simple and effective strategy is to eliminate objects with missing values.
- Eliminate attributes that have missing values.
- DataFrame.isna().sum() and DataFrame.isnull().sum() are used to find missing values in pandas.

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.
- There are several strategies for dealing with missing data:

Eliminate Data Objects or Attributes

- A simple and effective strategy is to eliminate objects with missing values.
- Eliminate attributes that have missing values.
- DataFrame.isna().sum() and DataFrame.isnull().sum() are used to find missing values in pandas.
- DataFrame.dropna() from Panda package in python can be used to eliminate data object with missing values. Included parameters:

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.
- There are several strategies for dealing with missing data:

Eliminate Data Objects or Attributes

- A simple and effective strategy is to eliminate objects with missing values.
- Eliminate attributes that have missing values.
- DataFrame.isna().sum() and DataFrame.isnull().sum() are used to find missing values in pandas.
- DataFrame.dropna() from Panda package in python can be used to eliminate data object with missing values. Included parameters:
  - **axis** :{0 or 'index' 1 or 'column'}, default 0

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.
- There are several strategies for dealing with missing data:

Eliminate Data Objects or Attributes

- A simple and effective strategy is to eliminate objects with missing values.
- Eliminate attributes that have missing values.
- DataFrame.isna().sum() and DataFrame.isnull().sum() are used to find missing values in pandas.
- DataFrame.dropna() from Panda package in python can be used to eliminate data object with missing values. Included parameters:
  - **axis** :{0 or 'index' 1 or 'column'}, default 0
  - **how** : {'any', 'all'} any: if any NA values are present, drop that row or column. All: if all values are NA, drop that row or column.

Missing Data

- In some cases, the information was not collected: e.g., some people decline to give their age or weight.
- There are several strategies for dealing with missing data:

Eliminate Data Objects or Attributes

- A simple and effective strategy is to eliminate objects with missing values.
- Eliminate attributes that have missing values.
- DataFrame.isna().sum() and DataFrame.isnull().sum() are used to find missing values in pandas.
- DataFrame.dropna() from Panda package in python can be used to eliminate data object with missing values. Included parameters:
  - **axis** :{0 or 'index' 1 or 'column'}, default 0
  - **how** : {'any', 'all'} any: if any NA values are present, drop that row or column. All: if all values are NA, drop that row or column.
  - **inplace**: If True, do operation and return None.

Sample Dataframe-Python

- Detecting missing values using df.isna()

```
import pandas as pd
import numpy as np

data = {"Product_Name":["Mouse", "Monitor", "CPU", "Speakers","Headset"],
        "Unit_Price":[200, 5000.235, 10000.550,  250.50, None],
        "No_Of_Units":[5, 10, 20,  8, pd.NaT],
        "Available_Quantity":[6,5,5, pd.NaT,np.NaN],
        "Remarks":[np.NaN,pd.NaT,pd.NaT,pd.NaT,pd.NaT]
       }
df = pd.DataFrame(data)

print(df.isna().sum())

df_copy1 = df.copy()
df_copy2 = df.copy()
df_copy3 = df.copy()

df_copy1.dropna(axis=1,inplace=True)
df_copy2.dropna(how='any',axis=1,inplace=True)
df_copy3.dropna(how='all',axis=1,inplace=True)
```

Missing Data

Estimate Missing values

- In time series dataset, we can not simply eliminate missing values.

Missing Data

Estimate Missing values

- In time series dataset, we can not simply eliminate missing values.
- Missing values in time series dataset can be estimated interpolated by using remaining values.

Missing Data

Estimate Missing values

- In time series dataset, we can not simply eliminate missing values.
- Missing values in time series dataset can be estimated interpolated by using remaining values.
- In dataset that has many similar data points, the attribute values of the points closest to the missing point can be used.

Missing Data

Estimate Missing values

- In time series dataset, we can not simply eliminate missing values.
- Missing values in time series dataset can be estimated interpolated by using remaining values.
- In dataset that has many similar data points, the attribute values of the points closest to the missing point can be used.
- If the attribute is continuous, then the average attribute values of the nearest neighborhood is used.
- If the attribute is categorical, then the most commonly occurring attribute value can be taken.

Missing Data

Estimate Missing values

- In time series dataset, we can not simply eliminate missing values.
- Missing values in time series dataset can be estimated interpolated by using remaining values.
- In dataset that has many similar data points, the attribute values of the points closest to the missing point can be used.
- If the attribute is continuous, then the average attribute values of the nearest neighborhood is used.
- If the attribute is categorical, then the most commonly occurring attribute value can be taken.
- DataFrame.fillna() is used to fill missing values using the specified method.

Missing Data

Estimate Missing values

- In time series dataset, we can not simply eliminate missing values.
- Missing values in time series dataset can be estimated interpolated by using remaining values.
- In dataset that has many similar data points, the attribute values of the points closest to the missing point can be used.
- If the attribute is continuous, then the average attribute values of the nearest neighborhood is used.
- If the attribute is categorical, then the most commonly occurring attribute value can be taken.
- DataFrame.fillna() is used to fill missing values using the specified method.
- DataFrame.replace(): can be used for replacement.

Sample Dataframe-Python

- Replacing missing values with specified integer.

```
# importing libraries
import pandas as pd
import numpy as np

nums = {'Set_of_Numbers': [2, 3, 5, 7, 11, 13,
                           np.nan, 19, 23, np.nan]}

# Create the dataframe
df1 = pd.DataFrame(nums, columns=['Set_of_Numbers'])
df2 = df1.copy()
# Apply the function
df2['Set_of_Numbers'] = df1['Set_of_Numbers'].fillna(0)

# print the DataFrame
print(df2)
```

Sample Dataframe-Python

- Replacing missing values mean and mode.

```
# importing pandas module
import pandas as pd

# making data frame from csv file
nba = pd.read_csv("C:\GW\Time series Analysis\dataset/nba.csv")
# check if missing value exists
print(nba.isna().sum())
# replacing na values in college with No college
nba["College"].fillna("No College", inplace = True)

# replacing missing values of 'College' feature
# with the mode of the same feature
nba["College"].fillna(nba["College"].mode()[0], inplace = True)

# replacing missing values of the 'Salary' with the mean value
# of the column
nba["Salary"].fillna(nba["Salary"].mean(), inplace = True)
```

Backward fill-Python

- Replacing missing values with backward fill option Dataframe.bfill().

```
import pandas as pd

# Creating a dataframe with "na" values.

df1 = pd.DataFrame({"A": [None, 1, 2, 3, None, None],
                    "B": [11, 5, None, None, None, 8],
                    "C": [None, 5, 10, 11, None, 8]})
df2 = df1.copy()
df3 = df1.copy()
# Filling missing values backward across row
df2.bfill(axis ='rows', inplace=True)
# Filling missing values backward across column
df3.bfill(axis ='columns', inplace=True)
```

Backward fill-Python

- Replacing missing values with backward fill option Dataframe.bfill().
- When axis='rows', then value in current na cells are filled from the corresponding value in the next row. If the next row is also na value then it won't be populated.

```
import pandas as pd

# Creating a dataframe with "na" values.

df1 = pd.DataFrame({"A": [None, 1, 2, 3, None, None],
                    "B": [11, 5, None, None, None, 8],
                    "C": [None, 5, 10, 11, None, 8]})
df2 = df1.copy()
df3 = df1.copy()
# Filling missing values backward across row
df2.bfill(axis ='rows', inplace=True)
# Filling missing values backward across column
df3.bfill(axis ='columns', inplace=True)
```

Backward fill-Python

- Replacing missing values with backward fill option Dataframe.bfill().
- When axis='rows', then value in current na cells are filled from the corresponding value in the next row. If the next row is also na value then it won't be populated.
- When axis='columns', then the current na cells will be filled from the value present in the next column in the same row. If the next column is also na cell then it won't be filled.

```python
import pandas as pd

# Creating a dataframe with "na" values.

df1 = pd.DataFrame({"A": [None, 1, 2, 3, None, None],
                    "B": [11, 5, None, None, None, 8],
                    "C": [None, 5, 10, 11, None, 8]})
df2 = df1.copy()
df3 = df1.copy()
# Filling missing values backward across row
df2.bfill(axis ='rows', inplace=True)
# Filling missing values backward across column
df3.bfill(axis ='columns', inplace=True)
```

Forward fill-Python

- Replacing missing values with forward fill option Dataframe.ffill().

```
# importing pandas as pd
import pandas as pd

# Creating the dataframe
df = pd.DataFrame({"A": [5, 3, None, 4],
                   "B": [None, 2, 4, 3],
                   "C": [4, 3, 8, 5],
                   "D": [5, 4, 2, None]})
df1 = df.copy()
df2 = df.copy()
# applying ffill() method to fill the missing values-row
df1.ffill(axis = 0, inplace=True)
# applying ffill() method to fill the missing values-column
df2 = df2.ffill(axis = 1)
```

Forward fill-Python

- Replacing missing values with forward fill option Dataframe.ffill().
- When ffill() is applied across the index, any missing value is filled based on the corresponding value in the previous row.

```
# importing pandas as pd
import pandas as pd

# Creating the dataframe
df = pd.DataFrame({"A": [5, 3, None, 4],
                   "B": [None, 2, 4, 3],
                   "C": [4, 3, 8, 5],
                   "D": [5, 4, 2, None]})
df1 = df.copy()
df2 = df.copy()
# applying ffill() method to fill the missing values-row
df1.ffill(axis = 0, inplace=True)
# applying ffill() method to fill the missing values-column
df2 = df2.ffill(axis = 1)
```

Forward fill-Python

- Replacing missing values with forward fill option Dataframe.ffill().
- When ffill() is applied across the index, any missing value is filled based on the corresponding value in the previous row.
- When ffill() is applied across the column axis, missing values are filled by the value in previous column in the same row.

```python
# importing pandas as pd
import pandas as pd

# Creating the dataframe
df = pd.DataFrame({"A": [5, 3, None, 4],
                   "B": [None, 2, 4, 3],
                   "C": [4, 3, 8, 5],
                   "D": [5, 4, 2, None]})
df1 = df.copy()
df2 = df.copy()
# applying ffill() method to fill the missing values-row
df1.ffill(axis = 0, inplace=True)
# applying ffill() method to fill the missing values-column
df2 = df2.ffill(axis = 1)
```

Forward & Backward fill-Python

- Forward and Backward fill can be applied using Dataframe.fillna() with method parameters to be set as:

```
import pandas as pd

# Creating a dataframe with "na" values.

df1 = pd.DataFrame({"A": [None, 1, 2, 3, None, None],
                    "B": [11, 5, None, None, None, 8],
                    "C": [None, 5, 10, 11, None, 8]})
df2 = df1.copy()
df3 = df1.copy()
df4 = df1.copy()
df5 = df1.copy()
# Filling missing values backward across row
df2.fillna(axis ='rows', method = 'bfill',inplace=True)
df3.fillna(axis ='columns', method = 'bfill',inplace=True)
# Filling missing values forward across column
df4.fillna(axis ='rows', method = 'ffill',inplace=True)
df5.fillna(axis ='columns', method = 'ffill',inplace=True)
```

Forward & Backward fill-Python

- Forward and Backward fill can be applied using Dataframe.fillna() with method parameters to be set as:
  - 'bfill/'backfill'

```
import pandas as pd

# Creating a dataframe with "na" values.

df1 = pd.DataFrame({"A": [None, 1, 2, 3, None, None],
                    "B": [11, 5, None, None, None, 8],
                    "C": [None, 5, 10, 11, None, 8]})
df2 = df1.copy()
df3 = df1.copy()
df4 = df1.copy()
df5 = df1.copy()
# Filling missing values backward across row
df2.fillna(axis ='rows', method = 'bfill',inplace=True)
df3.fillna(axis ='columns', method = 'bfill',inplace=True)
# Filling missing values forward across column
df4.fillna(axis ='rows', method = 'ffill',inplace=True)
df5.fillna(axis ='columns', method = 'ffill',inplace=True)
```

Forward & Backward fill-Python

- Forward and Backward fill can be applied using Dataframe.fillna() with method parameters to be set as:
  - 'bfill/'backfill'
  - 'ffill'/'pad'

```
import pandas as pd

# Creating a dataframe with "na" values.

df1 = pd.DataFrame({"A": [None, 1, 2, 3, None, None],
                    "B": [11, 5, None, None, None, 8],
                    "C": [None, 5, 10, 11, None, 8]})
df2 = df1.copy()
df3 = df1.copy()
df4 = df1.copy()
df5 = df1.copy()
# Filling missing values backward across row
df2.fillna(axis ='rows', method = 'bfill',inplace=True)
df3.fillna(axis ='columns', method = 'bfill',inplace=True)
# Filling missing values forward across column
df4.fillna(axis ='rows', method = 'ffill',inplace=True)
df5.fillna(axis ='columns', method = 'ffill',inplace=True)
```