3 Big Idea 2: Data

"Information is the oil of the 21st century, and analytics is the combustion engine."

—Peter Sondergaard, senior vice president, Gartner

Chapter Goals

- Using bits to represent data
- Abstractions
- Analog vs. digital data
- Consequences of using bits
- Number systems
- Converting numbers
- Overflow errors
- Roundoff errors
- Data compression
- Information from data
- Predicting algorithms
- Visualization of data
- Privacy concerns
- Metadata

Bits Represent Data

A bit is shorthand for a single binary digit and is either 0 or 1. A byte is 8 bits. For example, the binary sequence 0110 1111 contains 8 bits or 1 byte.

Binary sequences can be used to represent all digital data. Binary sequences can represent colors, Boolean logic, lists, and so on. Anything that can be stored on a computer can be represented by binary sequences.

Some data take many bits to represent it. For example, a single 10 MP (1 MP is one million pixels) picture (16-bit mode) uses 10,000,000 pixels. Each pixel contains 6 bytes. Each byte contains 8 bits. That means there are 8 bits * 6 * 10,000,000 = 480,000,000 bits in a single 10 MP (16-bit mode) picture. It is common for videos to use 1,000,000 bits per second of video.

Abstractions

Bits are grouped to represent abstractions. These abstractions include but are not limited to numbers, characters, and colors. Abstractions find common features to generalize the program. This can also help shrink the code if you are planning to use a method/procedure more than once in a program. Instead of repeating the code lines, you can reference a prior set of directions to repeat the outcome without having to rewrite the lines of code. By reducing the number of lines of code, chances for errors are also reduced.

Example One

When using the upper-level language JAVA, adding the numbers 1,234 + 4,321 would look like the following:

```
int x = 1234 + 4321
```

When using the upper level language Python, adding the number 1,234 + 4,321 would look like the following:

```
x = 1234 + 4321
```

Without abstractions and using just machine code, the same math example would look like the binary sequence below:

```
10111001 11010010 0000011 100001001 00001110 00000000 00000000
10111001 11100001 00010000 10001001 00001110 00000010 00000000
10100001 00000000 00000000 1001011 00011110 00000010 00000000
00000011 11000011 10100011 00000100 00000000
```

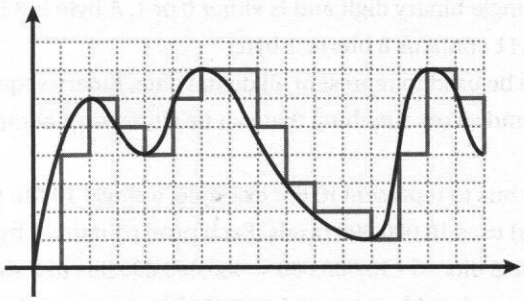
High-level languages contain the most abstractions and allow for easier coding and easier debugging.

Abstractions will be covered in more detail in the programming chapter.

Analog vs. Digital Data

An analog signal has values that change smoothly over time, rather than in discrete intervals. Some examples of analog data include pitch, the volume of music, the colors in a painting, or the position of a sprinter during a race. Analog signals are continuous signals, while digital signals are discrete time signals.

A digital signal is an analog signal that has been broken up into steps. Analog data can closely approximate digital data using a sampling technique, which means measuring values of the analog signal at regular intervals called samples. The samples are measured to figure out the exact bits required to store each sample. The smaller the sample rate, the more accurately the digital signal represents the analog signal.



Analog vs. digital signals

The use of digital data to approximate real-world analog data is an example of an abstraction.

Consequences of Using Bits to Represent Data

A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time. However, value can be a list or other collection that, in turn, contains multiple values. Some data types include integers, real numbers, Boolean, string, and list.

Examples:

Integer	4
Real Number	4.00
Boolean	True or False
String	"Novack the third"
List	[1, 1, 35, 6]

In many programming languages, integers are represented by a fixed number of bits, which limit the range of integer values and mathematical operations on those values. For example in JAVA, the range of the value of an integer is from −2,147,483,648 to +2,147,483,647. Trying to store a number bigger than the limits will result in an overflow error. Some languages like Python do not have limits on number size but, instead, expand to the limit of the available memory.

The language used on the AP exam, similar to the language Python, does not have a limit on the size of numbers. On the exam, the size of representable integers is limited only by the size of the computer's memory.

Although your test will not ask you what the limit is for a JAVA integer, the test will expect you to know that some computer languages do have limits on the size of data types.

Number Systems

Number bases, including binary, decimal, and hexadecimal, are used to represent and investigate digital data. On your AP exam, you will be expected to convert binary to decimal and decimal to binary only.

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Converting Numbers into Different Bases

Example Two

Convert a binary (BIN) number to a decimal (DEC) number.

$11011_{BIN} = ?_{DEC}$

Step 1. A five-column table is needed because 11011 has five digits. Start by putting a 1 into the upper-right box of the five-column table.

				1

Step 2. Fill in the remaining first row by continually multiplying by the base. Because the original number is in binary, fill the columns by continually multiplying the product by 2.

$8 \times 2 = 16$	$4 \times 2 = 8$	$2 \times 2 = 4$	$1 \times 2 = 2$	1

Step 3. Place the numbers to be converted into the second row.

16	8	4	2	1
1	1	0	1	1

Step 4. Add the result of multiplying row 1 by row 2.

$(16 * 1) + (8 * 1) + (4 * 0) + (2 * 1) + (1 * 1) = 27_{DEC}$

Answer: 27_{DEC}

Example Three

Convert a binary (BIN) number to a decimal (DEC) number.

$110_{BIN} = ?_{DEC}$

Step 1. Use a three-column table, and multiply the top row by the base number starting with 1 into the upper-right box.

4	2	1
1	1	0

Step 2. Add the result of multiplying row 1 by row 2.

$(4 * 1) + (2 * 1) + (1 * 0) = 6_{DEC}$

Answer: 6_{DEC}

Example Four

Convert a binary (BIN) number to a decimal (DEC) number.

$11011011_{BIN} = ?_{DEC}$

Step 1. Use an eight-column table, and multiply the top row by the base number starting with 1 in the upper-right box.

128	64	32	16	8	4	2	1
1	1	0	1	1	0	1	1

Step 2. Add the result of multiplying row 1 by row 2.

$(128 * 1) + (64 * 1) + (32 * 0) + (16 * 1) + (8 * 1) + (4 * 0) + (2 * 1) + (1 * 1) = 219_{DEC}$

Answer: 219_{DEC}

Example Five

Convert a hexadecimal (HEX) number to a decimal (DEC) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$12_{HEX} = ?_{DEC}$

Step 1. A two-column table is needed because 12 has two digits. Start by putting a 1 into the upper-right box of the two-column table.

	1

Step 2. Fill in the remaining first row by continually multiplying by the base. The base number is in HEX, so fill the columns by continually multiplying the product by 16.

1 × 16 = 16	1

Step 3. Place the numbers to be converted into the second row.

16	1
1	2

Step 4. Add the result of multiplying row 1 by row 2.

$(16 * 1) + (1 * 2) = 18_{DEC}$

Answer: 18_{DEC}

Decimal (base 10) goes from 0 to 9, which contains 10 numbers.
Binary (base 2) goes from 0 to 1, which contains two numbers.
Hexadecimal (base 16) goes from 0 to 15, which contains 16 numbers. To represent the numbers 10 through 15, the letters A to F are used.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
										A	B	C	D	E	F

Example Six

Convert a hexadecimal (HEX) number to a decimal (DEC) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$2B_{HEX} = ?_{DEC}$

Step 1. Use a two-column table, and multiply the top row by the base number starting with 1 in the upper-right box.

16	1
2	11

Step 2. Add the result of multiplying row 1 by row 2.

$(16 * 2) + (1 * 11) = 43_{DEC}$

Answer: 43_{DEC}

Example Seven

Convert a hexadecimal (HEX) number to a decimal (DEC) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$AD_{HEX} = ?_{DEC}$

Step 1. Use a two-column table, and multiply the top row by the base number starting with 1 in the upper-right box.

16	1
10	13

Step 2. Add the result of multiplying row 1 by row 2.

$(16 * 10) + (1 * 13) = 173_{DEC}$

Answer: 173_{DEC}

Example Eight

Convert an octal (OCT) number to a decimal (DEC) number.
(Note that octal (base 8) does not appear on the AP exam and is just included here for completeness.)

$28_{OCT} = ?_{DEC}$

Step 1. A two-column table is needed because 28 has two numbers. Start by putting a 1 into the upper-right box of the two-column table.

	1

Step 2. Fill in the remaining first row by continually multiplying by the base. Since the base number is in octal, fill the columns by continually multiplying the product by 8.

$1 \times 8 = 8$	1

Step 3. Place the numbers to be converted into the second row.

8	1
2	8

Step 4. Add the result of multiplying row 1 by row 2.

$(8 * 2) + (1 * 8) = 24_{DEC}$

Answer: 24_{DEC}

Example Nine

Convert an octal (OCT) number to a decimal (DEC) number.
(Note that octal (base 8) does not appear on the AP exam and is just included here for completeness.)

$127_{OCT} = ?_{DEC}$

Step 1. Use a three-column table, and multiply the top row by the base number starting with 1 in the upper-right box.

64	8	1
1	2	7

Step 2. Add the result of multiplying row 1 by row 2.

$(64 * 1) + (8 * 2) + (1 * 7) = 87_{DEC}$

Answer: 87_{DEC}

Example Ten

Convert a decimal (DEC) number to a binary (BIN) number.

$46_{DEC} = ?_{BIN}$

Step 1. Create a flexible table with enough columns until the number in the upper row is just bigger than the number you are converting. Since we are converting to binary, the table is in increments of 2^n (64 is bigger than 46).

64	32	16	8	4	2	1

Step 2. Start with the largest number that is still smaller than the target number. Subtract the number in the upper row of the table from the original number.

$46 - 32 = 14_{DEC}$

64	32	16	8	4	2	1
	1					

Step 3. Can 16 fit into 14? No, it cannot. So place a 0 under the 16 column.

64	32	16	8	4	2	1
	1	0				

Step 4. $14 - 0 = 14$; Can 8 fit into 14? Yes, it can.

$14 - 8 = 6$

64	32	16	8	4	2	1
	1	0	1			

Step 5. Can 4 fit into 6? Yes, it can.

$6 - 4 = 2$

64	32	16	8	4	2	1
	1	0	1	1		

Step 6. Can 2 fit into 2? Yes, it can.

$2 - 2 = 0$

64	32	16	8	4	2	1
	1	0	1	1	1	

Step 7. Can 1 fit into 0? No.

64	32	16	8	4	2	1
	1	0	1	1	1	0

Answer: 101110_{BIN}

Example Eleven

Convert a decimal (DEC) number to a binary (BIN) number.

$30_{DEC} = ?_{BIN}$

Step 1. Create a flexible table with enough columns until the number in the upper row is just bigger than the number you are converting.

32	16	8	4	2	1

Step 2. Start with the largest number that is still smaller than the target number. Subtract the number in the upper row of the table from the original number.

$30 - 16 = 14$

32	16	8	4	2	1
	1				

Step 3. $14 - 8 = 6$

32	16	8	4	2	1
	1	1			

Step 4. $6 - 4 = 2$

32	16	8	4	2	1
		1	1	1	

Step 5. $2 - 2 = 0$

32	16	8	4	2	1
	1	1	1	1	

Step 6: 0

32	16	8	4	2	1
	1	1	1	1	0

Answer: 11110_{BIN}

Example Twelve

What is the value in binary of $1111_{BIN} + 1_{DEC}$?

Use a table for all conversions. This problem can be solved in different ways. To keep the rules consistent, we will solve these types of problems the slow but steady way. This test gives 2 hours to solve 70 questions. We have time to solve this type of problem by converting the binary number to a decimal number, then adding the two decimal numbers, and then finally converting the decimal number back to a binary number.

8	4	2	1
1	1	1	1

$(8 * 1) + (4 * 1) + (2 * 1) + (1 * 1) = 8 + 4 + 2 + 1 = 15_{DEC}$

$15_{DEC} + 1_{DEC} = 16_{DEC}$

32	16	8	4	2	1
	1	0	0	0	0

$16 - 16 = 0$

Answer: 10000_{BIN}

Example Thirteen

What is the value in binary of $1001_{BIN} + 3_{DEC}$?

Use a table for all conversions. This problem can be solved in different ways. To keep the rules consistent, we will solve these types of problems the slow but steady way. Solve this type of problem by converting the binary number to a decimal number, then adding the two decimal numbers, and then finally converting the decimal number back to a binary number.

8	4	2	1
1	0	0	1

$(8 * 1) + (4 * 0) + (2 * 0) + (1 * 1) = 8 + 1 = 9_{DEC}$

$9_{DEC} + 3_{DEC} = 12_{DEC}$

16	8	4	2	1
	1	1	0	0

$12 - 8 = 4$

$4 - 4 = 0$

Answer: 1100_{BIN}

Example Fourteen

Convert a decimal (DEC) number to a hexadecimal (HEX) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$130_{DEC} = ?_{HEX}$

Step 1. Start with 1 in the upper right corner of a table. Continually multiple by 16 until the number in the upper row is just bigger than the number you are converting to. In this example 256 is just bigger then 130.

256	16	1

Step 2. 16 goes into 130 8 times with 2 remaining. $130 - (16 * 8) = 2$

256	16	1
	8	

Step 3. One goes into two twice with 0 remaining. $2 - (1 * 2) = 0$

256	16	1
	8	2

Answer: 82_{HEX}

Example Fifteen

Convert a decimal (DEC) number to a hexadecimal (HEX) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$163_{DEC} = ?_{HEX}$

Step 1. Create a flexible table with enough columns until the number in the upper row is just bigger than the number you are converting.

256	16	1

Step 2. $163 - (16 * 10) = 3$

256	16	1
	A	

Step 3. $3 - (1 * 3) = 0$

256	16	1
	A	3

Answer: $A3_{HEX}$

Binary numbers can be conveniently represented by hexadecimal numbers, where one hexadecimal digit represents four binary digits.

Example Sixteen

Convert a hexadecimal (HEX) number to a binary (BIN) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$BD_{HEX} = ?_{BIN}$

Step 1. Convert B (which equals 11_{DEC}) and D (which equals 13_{DEC}) separately.

B D
11 13

8	4	2	1		8	4	2	1
1	0	1	1		1	1	0	1

Step 2. Write the binary numbers one after the other, making sure to keep them in the correct order.

Answer: 10111101_{BIN}

Example Seventeen

Convert a hexadecimal (HEX) number to a binary (BIN) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$A9_{HEX} = ?_{BIN}$

Step 1. Convert A (which equals 10_{DEC}) and 9 separately.

10 9

8	4	2	1		8	4	2	1
1	0	1	0		1	0	0	1

Step 2. Write the binary numbers one after the other, making sure to keep them in the correct order.

Answer: 10101001_{BIN}

Example Eighteen

Convert a hexadecimal (HEX) number to a binary (BIN) number.
(Note that hexadecimal (base 16) does not appear on the AP exam and is just included here for completeness.)

$EE_{HEX} = ?_{BIN}$

Step 1. Convert E (which equals 15_{DEC}) twice.

15					15			
8	4	2	1		8	4	2	1
1	1	1	1		1	1	1	1

Step 2. Write the binary numbers one after the other, making sure to keep them in the correct order.

Answer: 11111111_{BIN}

Various Errors

Several different types of errors can occur, including overflow errors and roundoff errors.

Overflow Errors

In many programming languages, a fixed number of bits is used to represent characters or integer limits.

Example Nineteen

What is the largest value stored using 2 bits?

2	1

The largest number is 11_{BIN}.

2	1
1	1

$(2 * 1) + (1 * 1) = 3$

The formula to calculate the largest number stored is $2^n - 1$, where n is the number of bits.

$2^2 - 1 = 3$

The smallest number is 0.

Answer: So 2 bits can store numbers from 0 to 3, which is 4 numbers. If you try to store any number 4 or greater in a 2-bit system, you will get an overflow error.

Example Twenty

Using a 2-bit system, what would result from adding $5 + 3$?

Answer: To store the number 8 would require a 4-bit system, so you will get an overflow error. The largest number that can be stored in a 2-bit system is $2^2 - 1 = 3$.

Example Twenty-One

What is the largest value stored using 3 bits?

4	2	1
1	1	1

Largest number: $(4 * 1) + (2 * 1) + (1 * 1) = 7$

Using the formula gives $2^3 - 1 = 7$.

Smallest number: 0

Answer: So 3 bits can store numbers from 0 to 7, which is 8 numbers, or 2^3. If you try to store the number 8 or greater in a 3-bit system, you will get an overflow error.

Example Twenty-Two

What is the largest value stored using 4 bits?

8	4	2	1
1	1	1	1

Largest number: $(8 * 1) + (4 * 1) + (2 * 1) + (1 * 1) = 15$

Using the formula gives $2^4 - 1 = 15$.

Smallest number: 0

Answer: So 4 bits can store numbers from 0 to 15, which is 16 numbers, or 2^4.

Example Twenty-Three

What is the largest value stored using 8 bits?

Answer: The largest number is 255: $2^8 = 256$ numbers ranging from 0 to 255.

Notice that for every additional bit, the amount of numbers able to be stored is doubled.

Roundoff Errors

1/3 does not always equal 1/3.

A roundoff error occurs when decimals (real numbers) are rounded. One computer might calculate 1/3 as 0.333333. Another computer might calculate 1/3 as 0.3333333333. In this case, 1/3 on one computer is not equal to 1/3 on a second computer.

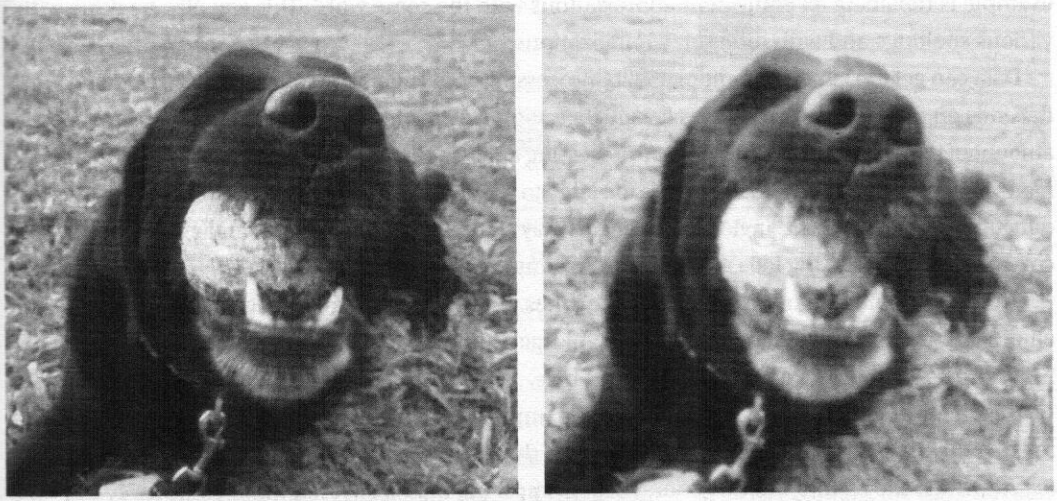
Lossy and Lossless Data Compression

Data compression is reducing the size (number of bits) of transmitted or stored data. Fewer bits does not necessarily mean less information.

Digital data compression often involves trade-offs in quality versus storage requirements.

Lossy compression can significantly reduce the file size while decreasing resolution. Traditionally, lossy compression is used to reduce file size for storage and transmission (email). The trade-off of using lossy data is that you will not recover the original file. Some data will be lost.

In lossless data compression, no data are lost. After compression, the original file can be reproduced without any lost data. The trade-off of lossless data compression is larger files that can be difficult to store, transfer, and handle.



Lossless (***left***) and lossy (***right***) data compression

In situations where quality or the ability to reconstruct the original file is important, lossless compression algorithms are typically chosen. In situations where minimizing data size or transmission time is important, lossy compression algorithms are typically chosen.

Information Extracted from Data

People generate significant amounts of digital data daily. Some always-on devices are collecting geographic location data constantly, while social media sites are collecting premium data based on your usage.

People can use computer programs to process information as well as to gain insight and knowledge. Information is the collection of facts and patterns extracted from data.

Gaining insight from this valuable data involves a combination of statistics, mathematics, programming, and problem solving. Large data sets may be analyzed computationally to reveal patterns, trends, and associations. These trends are powerful predictors of future behaviors. Investors are constantly reviewing trends in past pricing to influence their future investment decisions. However, sometimes trends can be misinterpreted and result in business disasters. Digitally processed data may show correlation between variables. A correlation found in data does not necessarily indicate that a causal relationship exists. Additional research is needed to understand the exact nature of the relationship.

Often, the size of the data set affects the amount of information that can be extracted from it. A single source often does not contain the data needed to draw a conclusion. Combining data from variety of sources may be necessary to formulate a conclusion.

Depending on how the data were collected, the information may not be uniform. For example, if users entered data into an open field, the way they chose to abbreviate, spell, or capitalize something may vary from user to user. Data sets pose challenges regardless of size, such as:

- The need to clean data
- Incomplete data
- Invalid data
- The need to combine data sources

Cleaning data is a process that makes the data uniform without changing their meaning. One example is replacing all equivalent abbreviations with the same word. This can also be done with various spellings and with different capitalizations.

Data can get too large for traditional data-processing applications. The ability to process data depends on the capabilities of the users and their tools. Social media activity generates an enormous amount of data. In the absence of a data-processing application, much of this data will go unexamined. All of the information in the data is too large to examine by hand in real time. Some data sets are difficult to process using a single computer and may require parallel systems. Parallel systems are fully covered in Chapter 5, "Big Idea 4: Computer Systems and Networks."

Problems of bias are often created by the types and sources of data being collected. Bias is not eliminated by simply collecting more data. A large amount of data is generated by humans. Algorithms that use this data will reflect this bias.

Despite the advantages of big data, a large sample size can magnify the bias associated with the data being used. Data can have little value if the sample is not representative of the population to which the results will be generalized. Computing bias is covered more completely in Chapter 6, "Big Idea 5: Impact of Computing."

Predicting Algorithms

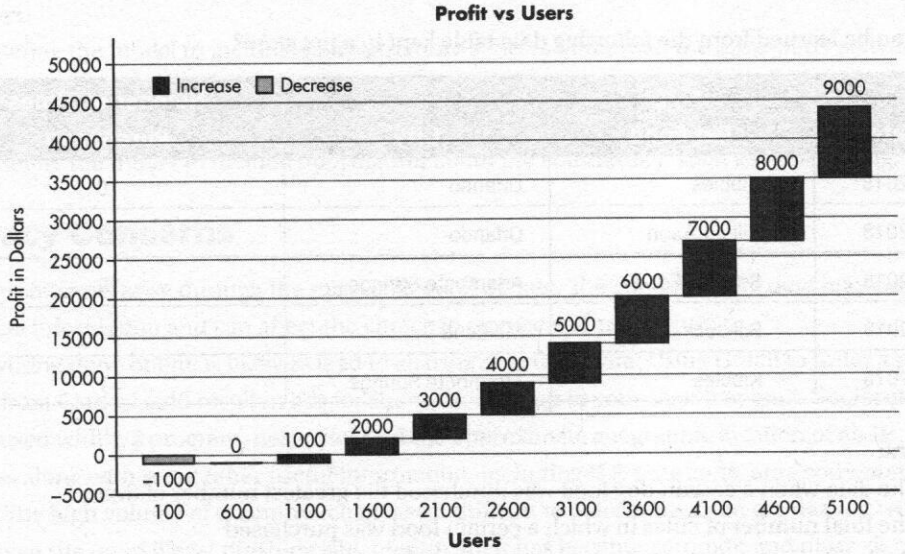
Predicting algorithms use information collected from big data to influence our daily lives. For example:

- A credit card company can use purchasing patterns to identify when to extend credit or flag a purchase for possible fraud.
- Social media sites can use patterns to target advertising based on viewing habits.
- An online store analyzing customers' past purchases can suggest new products the customer may be interested in buying.
- An entertainment application may recommend an additional movie to watch based on the viewer's interests.
- Algorithms can be used to prevent crimes by identifying crime "hot spots." The police can then step up patrols in those areas.

Visualization of Data

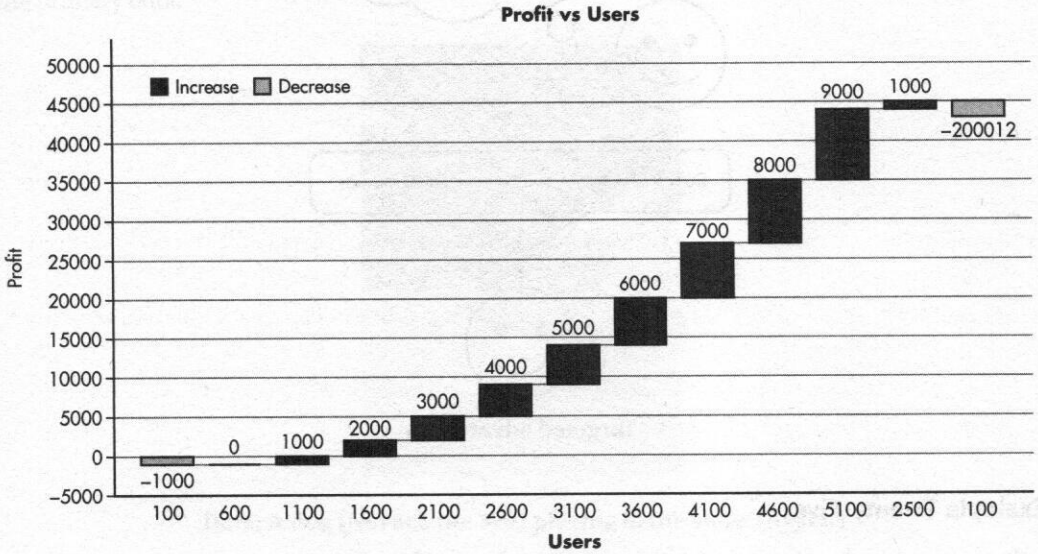
Using appropriate visualizations when presenting digitally processed data can help one gain insight and knowledge. Although big data is a powerful tool, the data will lose their value if they cannot be presented in a way that can be interpreted. Visualization tools can communicate information about data. Column charts, line graphs, pie charts, bar charts, XY charts, radar charts, histograms, and waterfall charts can make complex data easier to interpret.

For example, the graph below plots users versus profit. When looking at the trends from this graph, it looks like a direct relationship exists between the number of users and profit. The company might want to invest in drawing more members or spending on advertisers to draw in new members.



Graph of increasing profits

Predicting trends is not a guarantee of future usage. For example, the above graph cannot predict an innovation that could make this current innovation obsolete. It can be dangerous to draw conclusions based on good data and assume that those conclusions apply across the board or that past patterns will remain consistent. Often, a single source does not contain the data needed to draw a conclusion (see graph below). It may be necessary to combine data from a variety of sources to formulate a conclusion.



Graph of increasing profits with decrease at end of graph.

Predicting algorithms use historical data to predict future events. This data are used to build a mathematical model that encompasses trends. That predictive model is then used on current data to predict what will happen next.

Example Twenty-Four

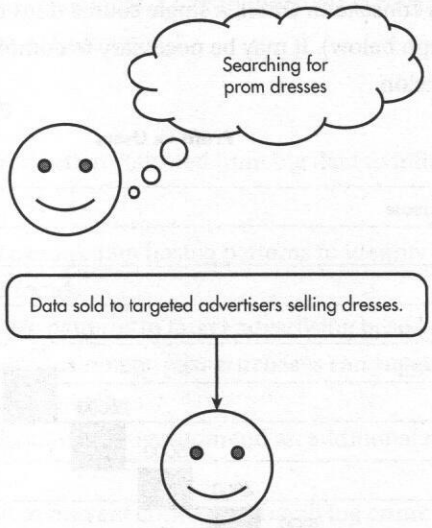
What can be learned from the following data table kept in a pet store?

Date	Pet Food	City	Number of Times Purchased
7/2018	Kibbles	Orlando	10
7/2018	Bill-Jackson	Orlando	2
7/2018	Science Food	Altamonte Springs	23
7/2018	Bill-Jackson	Maitland	37
7/2018	Kibbles	Altamonte Springs	1

Answers:

- The date when a certain dog food was purchased the greatest number of times
- The total number of cities in which a certain food was purchased
- The total number of foods purchased in a certain city during a month

E-commerce sites use data to determine how much inventory to hold and how to price products. Additionally, data about product views and purchases power the recommendation engine, which drives a large portion of sales. Data allow for personalized and effective advertisement. Sometimes an e-commerce site knows what you want to buy before you do.



Targeted advertisers

Example Twenty-Five

A high school principal is interested in predicting the number of students passing a state-level exam. She created a computer model that uses data from third-party software showing an increasing student pass rate for the exam. The model provided by the software company predicts a 90% student pass rate. The actual percentage of students passing the state exam was 74%.

When creating a model, all real-world variables cannot be represented. In the case of a model not accurately predicting outcomes, addition information can be added to make a more accurate prediction. What are some possible additions to the model to make it more reliable in predicting the student pass rate?

Answer:

- Refine the model to include data from more sources other than the third-party software due to the financial interest in the software being used.
- Refine the model to include student data from other schools.
- Refine the model to include information about the community, such as redistricting.

Privacy Concerns

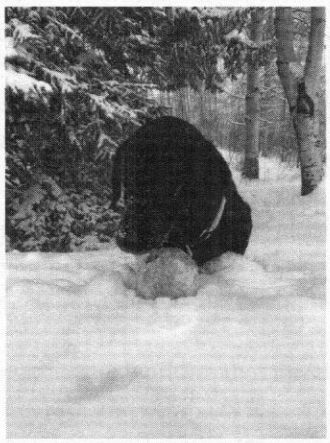
Privacy concerns arise through the mass collection of data. The content of the data may contain personal information and can affect the choice in storage and transmitting.

Anything done online is likely to lead to sharing of private data. Using Gmail to order a pair of shoes from Clarks could result in ads for shoes showing up in your search engine. Geolocation, when used within a program, helps you find the approximate geographic location of an IP address along with some other useful information, including ISP, time zone, area code, state, and so on. The high volume of e-commerce makes it difficult to determine if you are dealing with a legitimate site or an illegal phishing site. Identity theft has become common and more significant. The trade-off for the convenience of online shopping is the risk of violating privacy.

Metadata

Metadata are data that describe your data—for example, a picture of you standing in front of a waterfall is data. The location and time the picture was taken are metadata. Metadata are used for finding, organizing, and managing information.

Metadata can increase the effective use of data or data sets by providing additional information about various aspects of that data. Changes and deletions made to metadata do not change the primary data.



Data: A dog (Novack the 3rd) playing in the snow (Aspen)

A digital photo album contains metadata for each photo. The metadata are intended to help a search feature locate the popularity of geographic locations. All of the following are metadata and can be determined for the photo on page 147:

- The picture's filename
- The location where the picture was taken
- The date the picture was taken
- The author of the picture

Changes and deletions made to metadata do not change the primary data. Putting an image of Niagara Falls behind Novack the 3rd in the photo on page 147 will not change the location metadata of where the picture was taken.

By using metadata, pictures can be sorted based on where the picture was taken or sorted by the day the picture was taken. Metadata can be used to increase the effective use of data or data sets by providing this additional information.