- The first step in our data science process

scalability and maintainability in mind.

WebSocket services are also becoming more popular,

since they allow realtime notifications from the websites.

NoSQL storage systems are increasingly used to manage

a variety of data types.

These data stores are databases that do not represent data

in a table format with columns or rows,

as with conventional relational databases.

Examples of these data stores include cassandra,

mongoDB and HBASE.

NoSQL data stores provide APIs to allow users

to access the data.

These APIs can be used directly or in an application

that needs to access the data, like a Python script.

Additionally, most NoSQL systems provide data access

via a web interface, such as REST.

In one application from our work

at the San Diego Supercomputer Center,

we use wildfire data analysis to predict

fire direction and rate of spread.

This project requires acquiring data

using several different mechanisms.

The project itself stores historical sensor data

from weather stations in a relational database.

We use SQL to retrieve this data from the database

to create models to identify better patterns associated

with fire weather conditions.

To determine that a particular weather station

is currently experiencing fire weather conditions, we access

realtime data using a WebSocket service.

Once we start listening to the service, we receive

weather station measurements as the occur.

This data is then processed and compared to patterns found

by our models to determine if a better station

is experiencing Santa Ana conditions,

or fire weather conditions

At the same time, tweets are retrieved using hashtags

related to any fires that are occurring in the region.

The tweet messages are retrieved using

the Twitter REST service.

The idea is to determine the sentiment of these tweets

to see if people are expressing fear, anger

or simply nonchalant about the nearby fire.

The combination of censor data and tweet sentiments

helps to give us a sense of the urgency

of the fire situation.

To summarize, data can come from many places.

Finding and evaluating all useful data to our analytics

is important before we start acquiring data.

Depending on the source and structure of the data,

there are alternative ways to access it and we'll see all

about these access methods throughout

the coming weeks in Python.

is acquiring data.

After this video, you will be able to list

techniques and technologies to access and retrieve the data

you need, and describe an example scenario

that accesses data from a variety of sources

using different technologies.

As I said, the first step is acquire data.

That means you need to obtain the source material

before analyzing or acting on it.

The first step in acquiring data is determining

what data is available.

When it comes to finding the right data sources,

we try to leave no stone unturned.

You want to identify suitable data

related to your problem, and make use of all data

that is relevant to your problem for analysis.

Sometimes, leaving out even a small amount

of important data can lead to incorrect conclusions.

Data comes from many places, local and remote,

in many varieties, structured and unstructured,

and in many different velocities, that refers

to the streaming speed of the data.

There are many techniques and technologies to access these

different types of data.

Let's discuss a few examples.

A lot of data exists in conventional, relational databases,

like structured data coming from organizations.

The tool of choice access data from databases

is Structured Query Language, or SQL, which is supported

by all relational database management systems.

Additionally, most database systems come with a graphical

application environment that allows you to query and explore

the data sets in the database.

Data can also exist in files such as text files

and Excel spreadsheets.

Scripting languages are generally used to get data

from files.

A scripting language, like Python,

is a high level programming language that can be

general purpose or specialized for specific functions.

You'll see unique scripting and some Python in week two.

In addition, throughout this class, we will be using Python

in all of our case studies and examples.

Next week, we'll review Python and in week three,

you'll start learning Python libraries and functions

related to text processing.

Other common scripting languages with support

for processing files are JavaScript, php, Perl, R,

Octa and MATLAB to name a few.

An increasingly popular way to get data is from websites.

Webpages are written using a set of senders approved

by worldwide web consortium, or W3C.

This includes a variety of formats and services.

One common format is XML or Extensible Markup Language

or JSON, which both use markup symbols and tabs

to describe the contents on a webpage.

Many websites also host webs services

which provides programmatic access to their data.

There are several types of web services,

the most popular one being REST since it's easy to use.

REST stands for Representational State Transfer, and it is

an approach for implementing webs services with performance