Timeseries kinds and applications

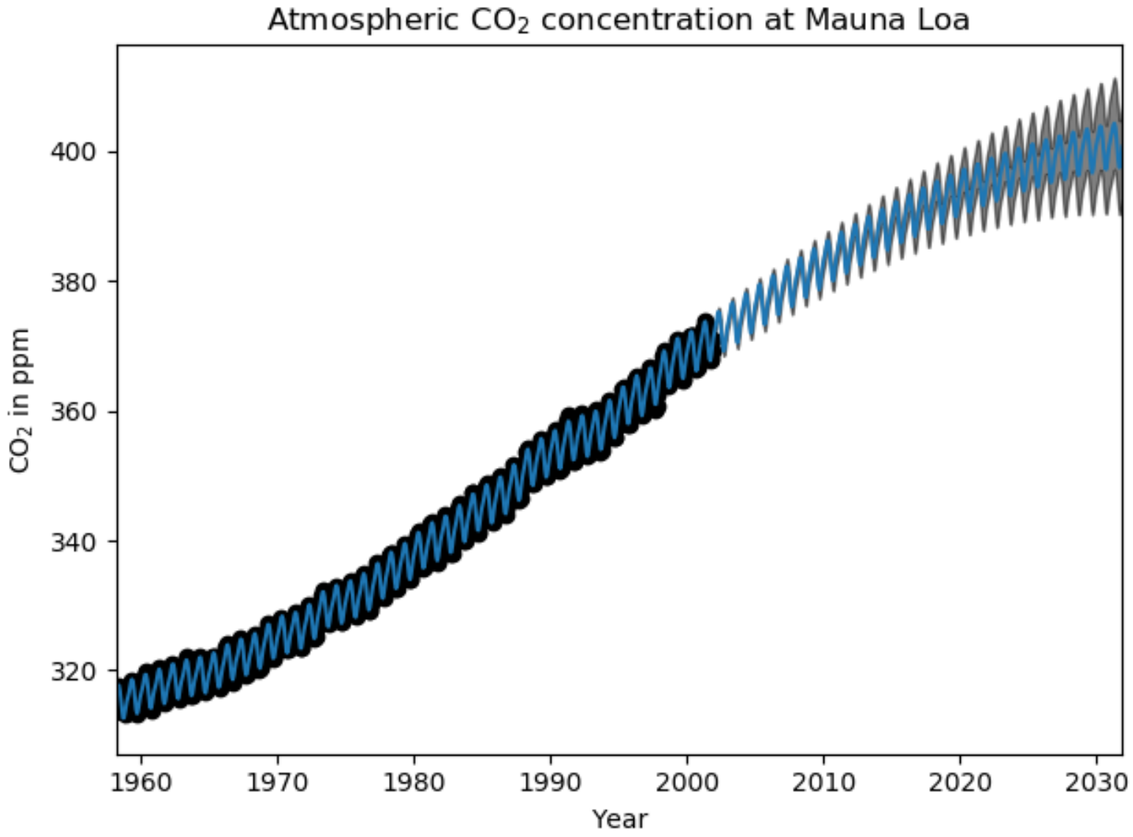
MACHINE LEARNING FOR TIME SERIES DATA IN PYTHON

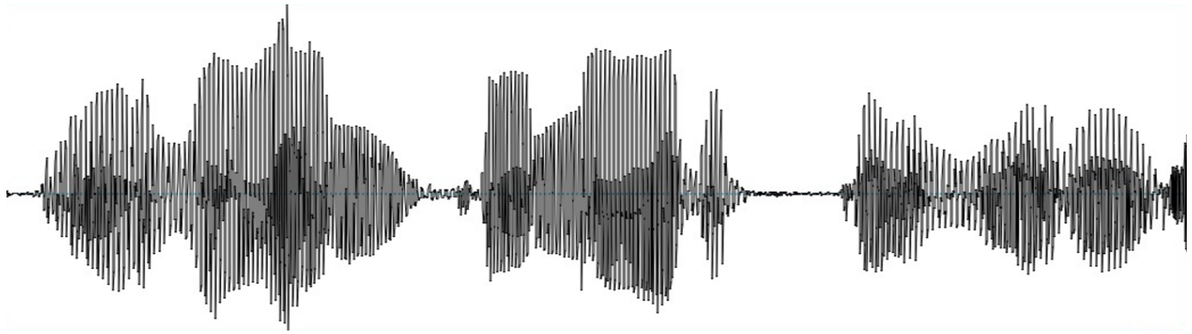
Chris Holdgraf

Fellow, Berkeley Institute for Data Science

datacamp

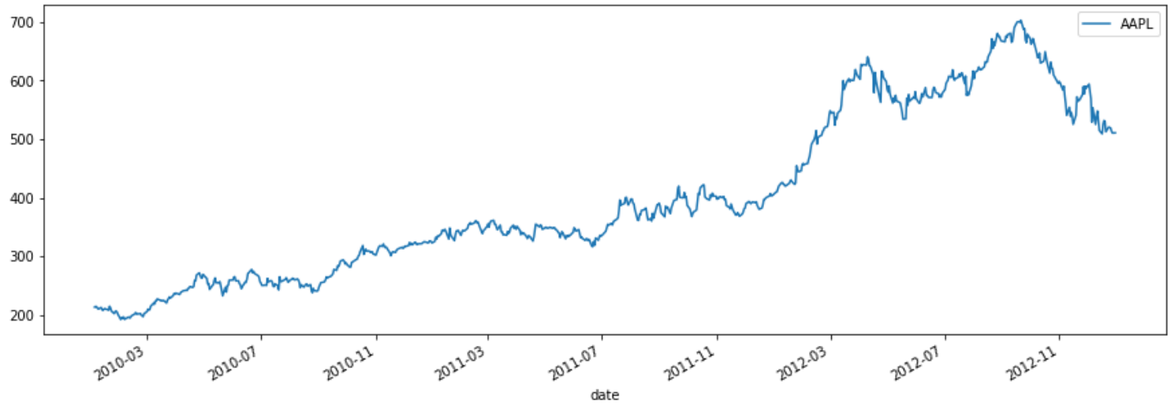
Time Series

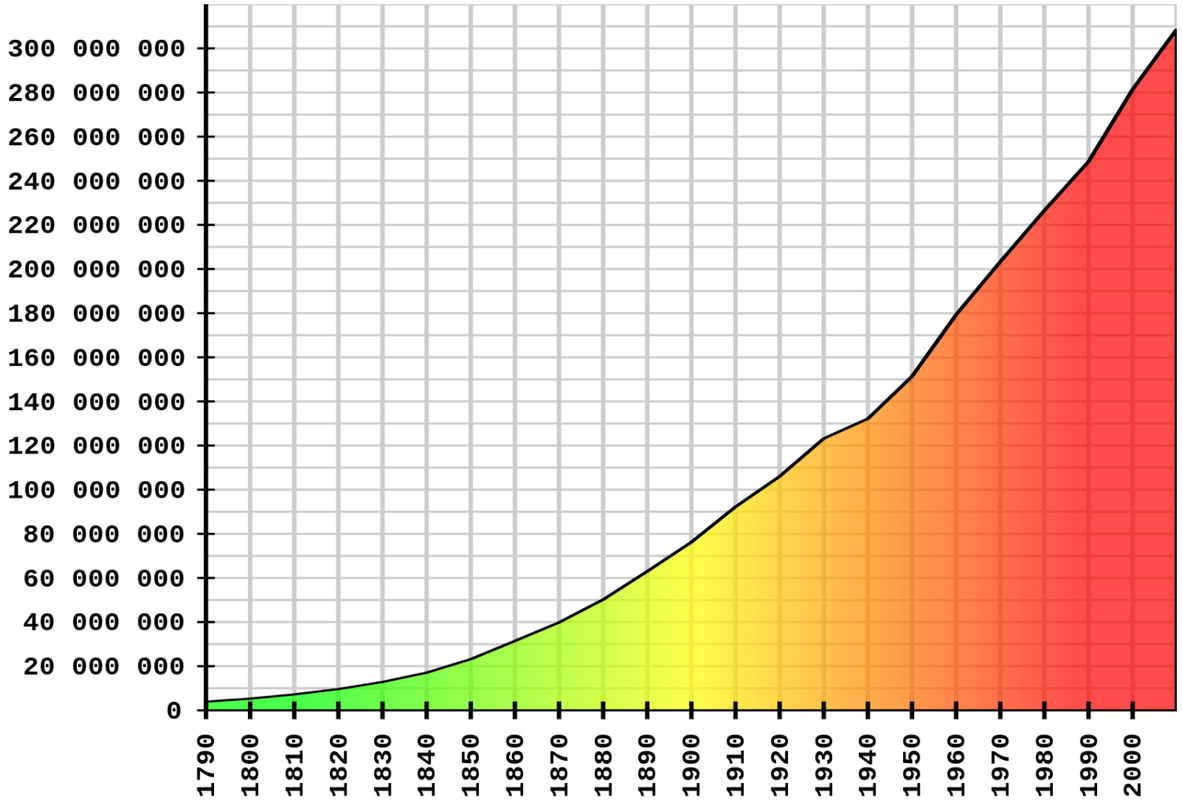




시계열은 시간이 지남에 따라 변화하는 데이터를 의미
이 페이지에서는 시간이 지남에 따라 발생하는 대기의 이산화탄소,
말하고 있는 목소리의 파형과 같은 다양한 형태를 취할 수 있음.

Time Series





한 해 동안 주식 가치의 변동 또는 도시에 대한 인구 통계 정보

What makes a time series?

Datapoint	Datapoint	Datapoint	Datapoint	Datapoint	Datapoint
1	34	12	54	76	40

Timepoint	Timepoint	Timepoint	Timepoint	Timepoint	Timepoint
2:00	2:01	2:02	2:03	2:04	2:05

Timepoint	Timepoint	Timepoint	Timepoint	Timepoint	Timepoint
Jan	Feb	March	April	May	Jun

Timepoint	Timepoint	Timepoint	Timepoint	Timepoint	Timepoint
1e-9	2e-9	3e-9	4e-9	5e-9	6e-9

시계열 데이터의 구성
1. 데이터 자체를 나타내는 숫자 배열
2. 각 데이터 지점에 대한 타임스탬프가 들어있는 또 다른 배열
(타임스탬프는 1년 중 수개월에서 나노초까지 다양한 범위의 시가니 데이터가 포함될 수 있음.)

Reading in a time series with Pandas

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('data.csv')
data.head()
```

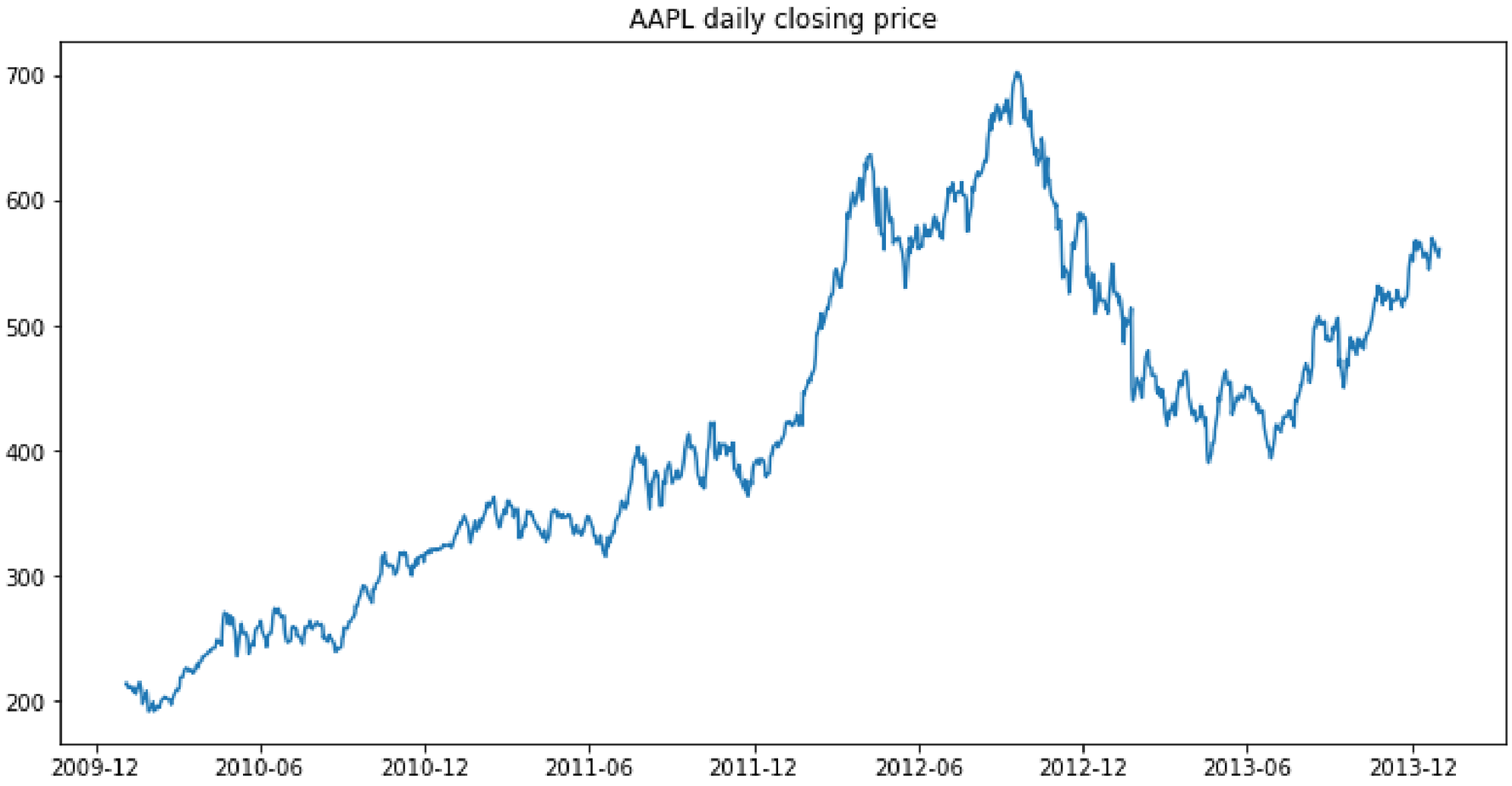
각 데이터 지점에는 동일한 시간대를 가질 수 있지만, 각 데이터 지점에는 해당하는 시간대가 있습니다(이 경우 날짜).

	date	symbol	close	volume
0	2010-01-04	AAPL	214.009998	123432400.0
46	2010-01-05	AAPL	214.379993	150476200.0
92	2010-01-06	AAPL	210.969995	138040000.0
138	2010-01-07	AAPL	210.580000	119282800.0
184	2010-01-08	AAPL	211.980005	111902700.0

Plotting a pandas timeseries

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(12, 6))
data.plot('date', 'close', ax=ax)
ax.set(title="AAPL daily closing price")
```

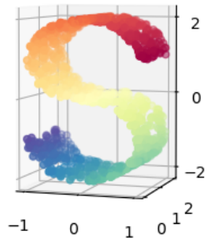
A timeseries plot

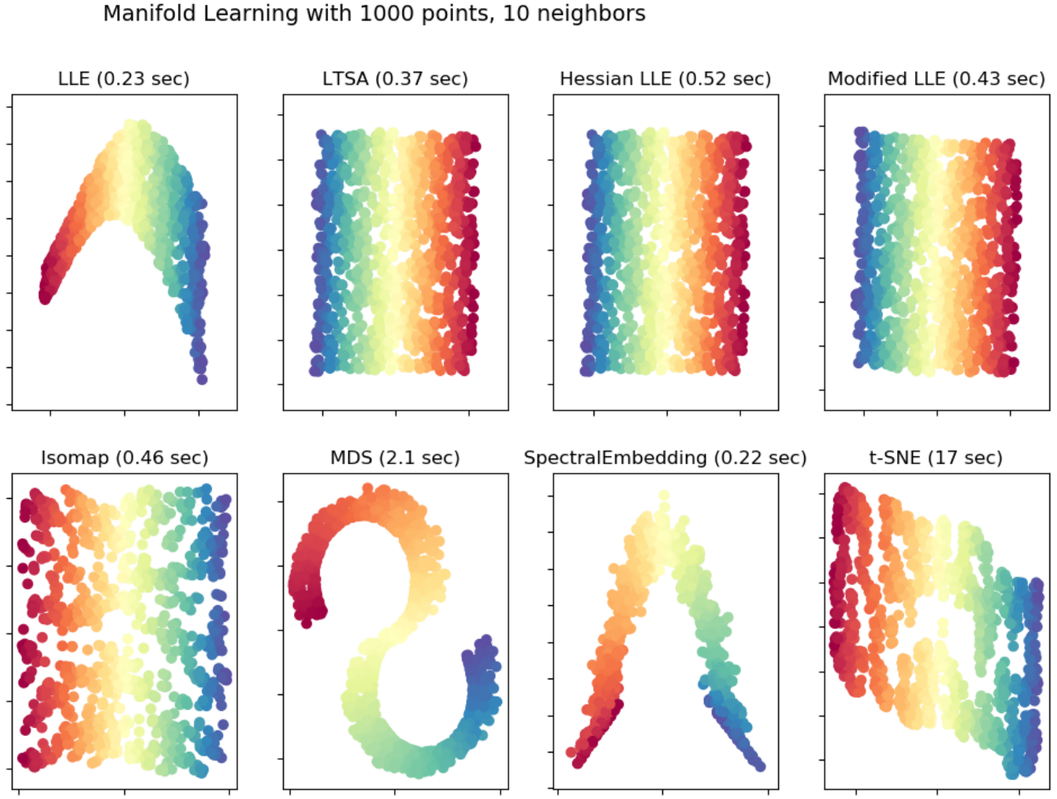


Why machine learning?

We can use really big data and really complicated data

기계 학습은 데이터 과학의 세계를 휩쓸었다. 지난 수십 년 동안, 컴퓨팅 능력, 알고리즘, 그리고 지역사회 관행의 발전은 컴퓨터를 사용하여 결코 가능하지 않다고 생각되었던 질문을 하는 것을 가능하게 했다. 기계 학습은 데이터에서 패턴을 찾는 것입니다. 종종 인간의 눈에는 바로 알 수 없는 패턴입니다. 이것은 종종 데이터가 너무 크거나 너무 복잡해서 사람이 처리할 수 없기 때문입니다.



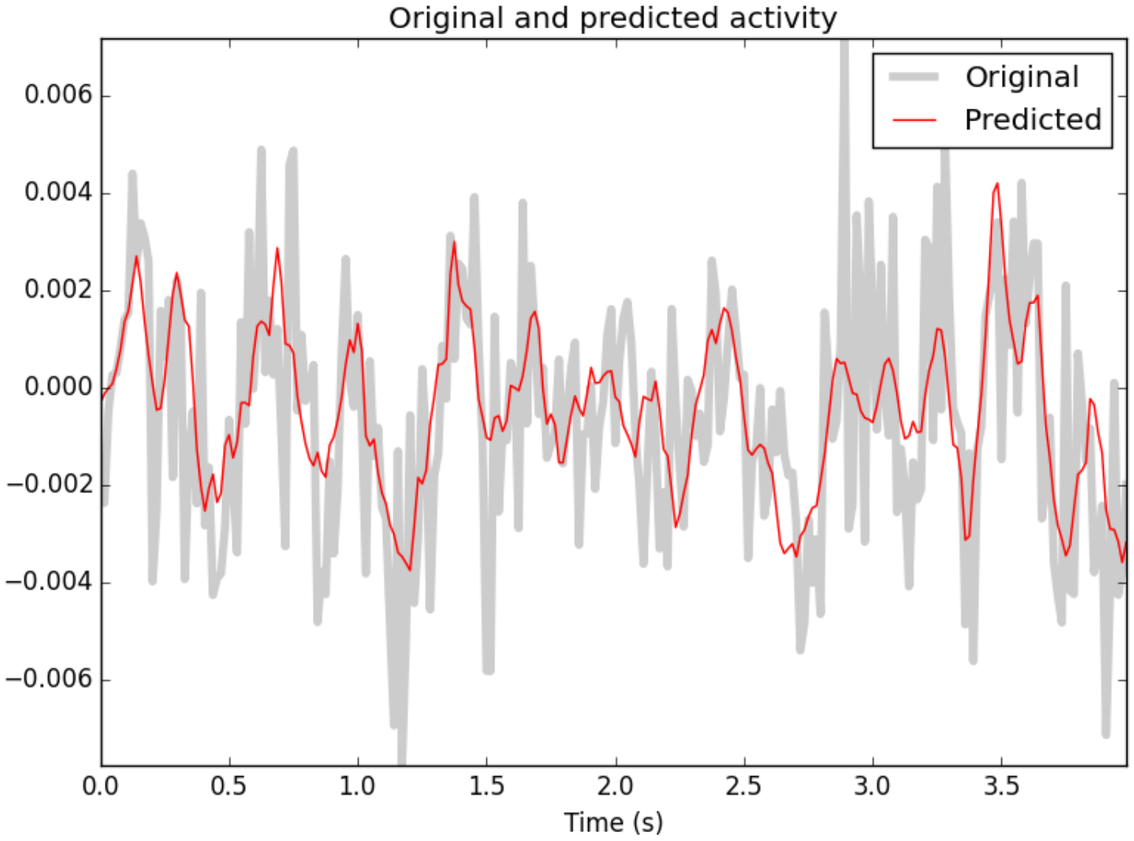


Why machine learning?

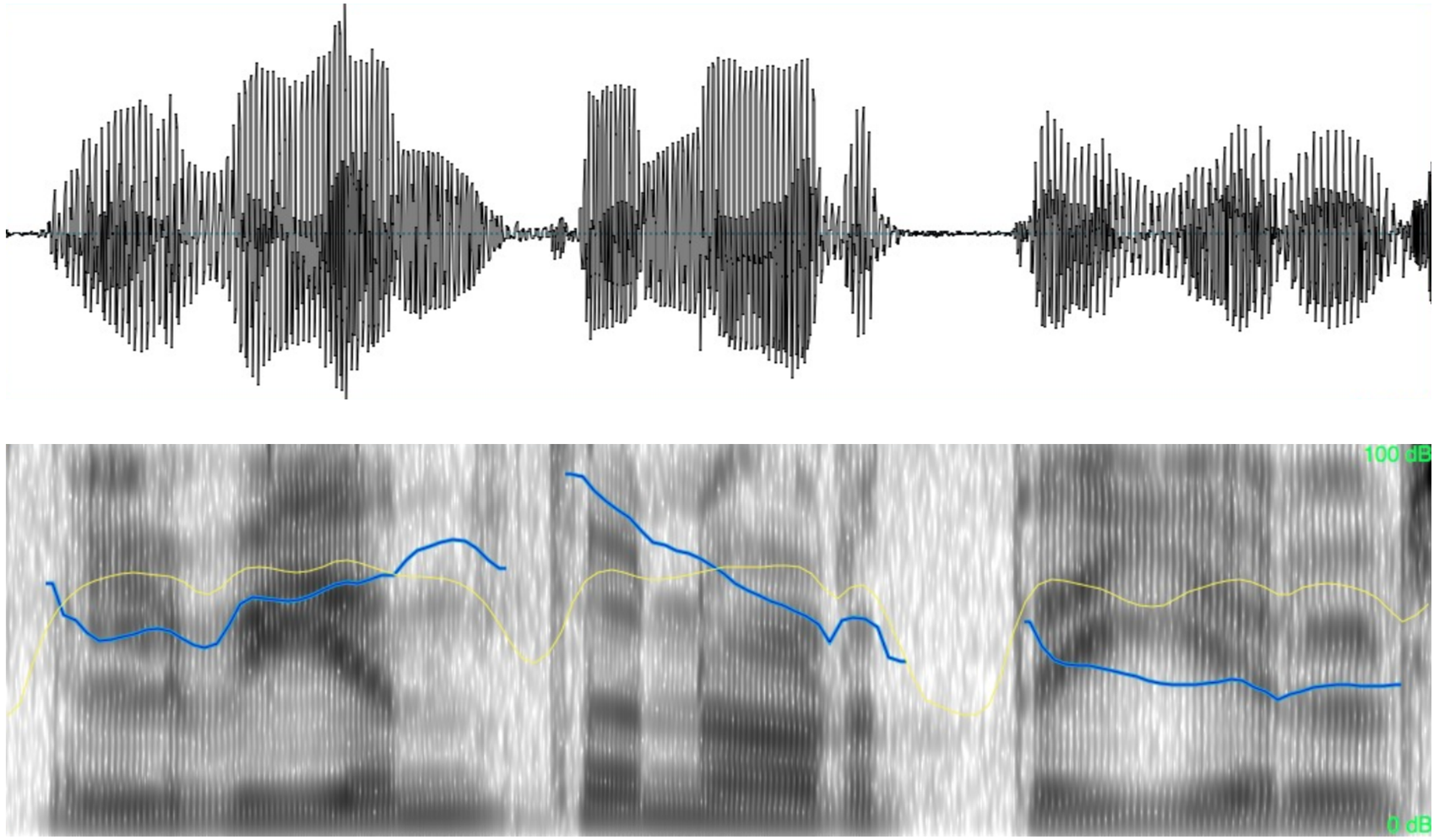
We can...

- Predict the future
- Automate this process

기계 학습의 또 다른 중요한 부분은 우리가 당면한 문제에 대한 우리의 지식을 공식화하는 세계의 모델을 구축할 수 있다는 것이다. 우리는 이 모델을 사용하여 예측을 할 수 있습니다. 자동화와 함께 이러한 기능은 조직의 의사 결정에서 중요한 구성 요소가 될 수 있습니다.



Why combine these two?



시계열을 다른 데이터 세트와 다르게 취급해야 하는 이유는 무엇입니까? 기계 학습은 데이터에서 패턴을 찾는 것입니다. 시계열 데이터는 시간이 지남에 따라 항상 변하므로 유용한 패턴으로 나타납니다. 예를 들어, 이것은 누군가가 말하는 원시 파형이고, 이것은 거기에서 추출된 시계열 피쳐 모음입니다. 보시다시피 시계열별 기능을 사용하면 원시 데이터를 훨씬 더 풍부하게 표현할 수 있습니다.

A machine learning pipeline

- Feature extraction
- Model fitting
- Prediction and validation

이 과정에서는 시계열 데이터의 맥락에서 간단한 기계 학습 파이프라인에 초점을 맞춘다. 이것은 다음과 같은 주요 단계로 요약된다. 특징 추출: 시간이 지남에 따라 변화하는 신호를 활용하는 특수 기능은 무엇입니까? 모델 피팅: 시계열 데이터를 사용하여 질문하는 데 적합한 모델은 무엇입니까? 유효성 검사: 시계열 데이터를 사용하는 모델을 검증하려면 어떻게 해야 합니까? 시간이 지나면 변하기 때문에 우리는 어떤 고려를 해야 할까요?

Let's practice!

MACHINE LEARNING FOR TIME SERIES DATA IN PYTHON

Machine learning basics

MACHINE LEARNING FOR TIME SERIES DATA IN PYTHON

Chris Holdgraf

Fellow, Berkeley Institute for Data Science

datacamp

Always begin by looking at your data

```
array.shape
```

```
(10, 5)
```

```
array[:3]
```

```
array([[ 0.735528  ,  1.00122818, -0.28315978],
       [-0.94478393,  0.18658748, -0.00241224],
       [-0.74822942, -1.46636618,  0.69835096]])
```

Always begin by looking at your data

```
df.head()
```

```
       col1      col2      col3
0  0.735528  1.001228 -0.283160
1 -0.944784  0.186587 -0.002412
2 -0.748229 -1.466366  0.698351
3  1.038589 -0.171248  0.831457
4 -0.161904  0.003972 -0.321933
```

Always visualize your data

Make sure it looks the way you'd expect.

```
# Using matplotlib
fig, ax = plt.subplots()
ax.plot(...)

# Using pandas
fig, ax = plt.subplots()
df.plot(..., ax=ax)
```

Scikit-learn

Scikit-learn is the most popular machine learning library in Python

```
from sklearn.svm import LinearSVC
```

Preparing data for scikit-learn

- `scikit-learn` expects a particular structure of data:

  `(samples, features)`

- Make sure that your data is *at least two-dimensional*
- Make sure the first dimension is *samples*

scikit - learn은 데이터가 2차원이어야함
(sample, feature)

If your data is not shaped properly

- If the axes are swapped:

```
array.T.shape 축을 바꿀 수 있음. (2차원일 때 가장 유용)
```

```
(10, 3)
```

If your data is not shaped properly

- If we're missing an axis, use `.reshape()` :

```
array.shape
```

```
(10,)
```

```
array.reshape([-1, 1]).shape
```

```
(10, 1)
```

- `-1` will automatically fill that axis with remaining values

데이터가 제대로 형성되지 않은 경우
또 다른 옵션은 원하는 모양을 지정할 수 있는 도트 재구성 방법을 사용

Fitting a model with scikit-learn

```
# Import a support vector classifier
from sklearn.svm import LinearSVC

# Instantiate this model
model = LinearSVC()

# Fit the model on some data
model.fit(X, y)
```

It is common for `y` to be of shape `(samples, 1)`

Investigating the model

```
# There is one coefficient per input feature
model.coef_
```

```
array([[ 0.69417875, -0.5289162 ]])
```

모형이 발견한 패턴의 종류를 조사하는 것이 종종 유용합니다. 대부분의 모델은 점 적합을 호출한 후 생성되는 속성에 이 정보를 저장합니다. 여기서는 모델이 각 형상에 부여한 계수를 보여준다.

Predicting with a fit model

```
# Generate predictions
predictions = model.predict(X_test)
```

모형이 적합하면 모형의 점 예측 방법을 호출하여 보이지 않는 데이터 점의 레이블을 결정할 수 있습니다.

Let's practice

MACHINE LEARNING FOR TIME SERIES DATA IN PYTHON

Combining timeseries data with machine learning

MACHINE LEARNING FOR TIME SERIES DATA IN PYTHON

Chris Holdgraf

Fellow, Berkeley Institute for Data Science

datacamp

Getting to know our data

- The datasets that we'll use in this course are all freely-available online
- There are many datasets available to download on the web, the ones we'll use come from Kaggle

먼저 사용할 데이터에 대해 간략하게 살펴보겠습니다. 둘 다 온라인에서 무료로 이용할 수 있고, 훌륭한 웹사이트인 캐글닷컴에서 왔습니다.

The Heartbeat Acoustic Data

- Many recordings of heart sounds from different patients
- Some had normally-functioning hearts, others had abnormalities
- Data comes in the form of audio files + labels for each file
- Can we find the "abnormal" heart beats?

오디오는 매우 일반적인 종류의 시계열 데이터입니다. 오디오는 샘플링 주파수가 매우 높은 경향이 있습니다(흔히 초당 20,000개 이상의 샘플). 우리의 첫 번째 데이터 세트는 의료 환자의 심장으로부터 기록된 오디오 데이터이다. 이 환자들 중 일부는 심장 이상을 가지고 있다. 이 심장 박동 데이터만 사용하여 어떤 피험자에게 이상이 있는지 알 수 있습니까?

Loading auditory data

```
from glob import glob
files = glob('data/heartbeat-sounds/files/*.wav')


print(files)
```

```
['data/heartbeat-sounds/proc/files/murmur__201101051104.wav',
 ...
 'data/heartbeat-sounds/proc/files/murmur__201101051114.wav']
```

오디오 데이터는 종종 "파형" 파일에 저장됩니다. "glob" 함수를 사용하여 이러한 파일을 모두 나열할 수 있습니다.
지정된 패턴과 일치하는 파일을 나열합니다.
각 파일에는 하나의 심장박동 세션에 대한 청각 데이터와 해당 데이터에 대한 샘플링 속도가 포함됩니다.

Reading in auditory data

```
import librosa as lr
# `load` accepts a path to an audio file
audio, sfreq = lr.load('data/heartbeat-sounds/proc/files/murmur__201101051104.wav')

print(sfreq)
```

```
2205
```

In this case, the sampling frequency is `2205` , meaning there are `2205` samples per second

"librosa"라는 라이브러리를 사용하여 오디오 데이터 집합을 읽습니다. Librosa는 청각 데이터에 대한 특징, 시각화 및 분석을 추출하는 기능을 가지고 있습니다. 우리는 "load" 기능을 사용하여 데이터를 가져올 수 있습니다. 데이터는 오디오에 저장되고 샘플링 주파수는 sfreq에 저장됩니다. 여기서 샘플링 주파수는 2205이며, 이는 초당 2205개의 샘플이 기록된다는 것을 의미합니다.

Inferring time from samples

- If we know the sampling rate of a timeseries, then we know the timestamp of each datapoint *relative to the first datapoint*
- Note: this assumes the sampling rate is fixed and no data points are lost

샘플링 주파수만 사용하여 파일의 시작에 상대적인 오디오 파일의 각 데이터 포인트의 시점을 유추할 수 있습니다.

Creating a time array (I)

- Create an array of indices, one for each sample, and divide by the sampling frequency

```
indices = np.arange(0, len(audio))
time = indices / sfreq
```

이제 데이터에 대한 타임스탬프 배열을 생성하겠습니다. 이렇게 하려면 두 가지 옵션이 있습니다. 첫 번째 방법은 오디오 파일의 0부터 데이터 포인트 수까지의 범위의 인덱스를 생성하고 각 인덱스를 샘플링 주파수로 나눈 다음 각 데이터 포인트에 대한 시점을 갖는 것입니다.

Creating a time array (II)

- Find the time stamp for the *N-1*th data point. Then use `linspace()` to interpolate from zero to that time

```
final_time = (len(audio) - 1) / sfreq
time = np.linspace(0, final_time, sfreq)
```

두 번째 옵션은 유사한 방법을 사용하여 오디오 데이터의 최종 시점을 계산하는 것입니다. 그런 다음 선형 공간 함수를 사용하여 0과 최종 시점 사이에 균등한 간격의 숫자를 생성합니다. 두 경우 모두 오디오 데이터와 같은 길이의 숫자 배열이 있어야 합니다.

The New York Stock Exchange dataset

- This dataset consists of company stock values for 10 years
- Can we detect any patterns in historical records that allow us to predict the value of companies in the future?

다음은 뉴욕 증권 거래소의 데이터를 살펴보겠습니다. 그것은 우리의 오디오 데이터보다 훨씬 긴 시간 동안 실행되며, 하루에 한 개의 샘플로 샘플링 주파수를 갖는다(오디오 데이터와 초당 2,205개의 샘플과 비교). 우리의 목표는 시장의 과거 데이터를 사용하여 회사의 주식 가치를 예측하는 것입니다. 연속 출력 값을 예측하고 있기 때문에, 이것은 회귀 문제입니다.

Looking at the data

```
data = pd.read_csv('path/to/data.csv')


data.columns
```

```
Index(['date', 'symbol', 'close', 'volume'], dtype='object')
```

```
data.head()
```

```
         date symbol       close       volume
0  2010-01-04   AAPL  214.009998  123432400.0
1  2010-01-04    ABT   54.459951   10829000.0
2  2010-01-04    AIG   29.889999    7750900.0
3  2010-01-04   AMAT   14.300000   18615100.0
4  2010-01-04   ARNC   16.650013   11512100.0
```

Timeseries with Pandas DataFrames

- We can investigate the object type of each column by accessing the `dtypes` attribute

```
df['date'].dtypes
```

```
0    object
1    object
2    object
dtype: object
```

Converting a column to a time series

- To ensure that a column within a DataFrame is treated as time series, use the `to_datetime()` function

```
df['date'] = pd.to_datetime(df['date'])


df['date']
```

하나의 열이 실제로 날짜 목록이라는 것을 알고 있으므로 to_datetime 함수를 사용하여 열 유형을 "datetime"으로 변경해 보겠습니다. 이렇게 하면 나중에 시각화 및 분석을 수행할 수 있습니다.

```
0   2017-01-01
1   2017-01-02
2   2017-01-03
Name: date, dtype: datetime64[ns]
```

Let's practice!

MACHINE LEARNING FOR TIME SERIES DATA IN PYTHON