

Data Science | 30 Days of Machine Learning | Day - 4

Educator Name: Nishant Dhote
Support Team: **+91-7880-113-112**

#Data Gathering

1. Working with CSV Files – Day 3
2. Working with JSON/SQL – Day 3

#Framing a Machine Learning Problem

#Data Gathering

3. Fetching data from an API
4. Fetching data using web scraping

→ Framing a Machine Learning Problem

Framing an ML problem

After verifying that your problem is best solved using either a predictive ML or a generative AI approach, you're ready to frame your problem in ML terms. You frame a problem in ML terms by completing the following tasks:

- Define the ideal outcome and the model's goal.
- Identify the model's output.
- Define success metrics.

Read Article: <https://developers.google.com/machine-learning/problem-framing/ml-framing>

1. Business problem to ML problem
2. Type of problem
3. Current solution
4. Getting data
5. Metrics to measure
6. Online Vs Batch
7. Check assumptions

Case Study Netflix – <https://tinyurl.com/4mscpz2t>

Step 1: Transaction Business problem to ML problem: Increases our revenue and decrease the churn rate.

Step 2: Identify the problem: Identify the customers who leave the platform.
Subscription Discount | UI Not Navigate | Content Demand | Internet Issue
Sentiment score

Step 3: Current solution: Overall churn rate prediction in last year review.

Step 4: Getting Data: We required exact date | Watch Time | Search Result
Matching | Show Publish Date | Recommendation Click | Left Show in Middle
Combine Discuss With - Data Engineering team & create a warehouse.

Step 5: Metrix to measure: Follow check points and rectify the direction of correct Metrix.

Step 6: Online Vs Batch ML: Focus on training model and retraining model.
OLTP system in data warehouse Setup: OLTP, or online transactional processing, enables the real-time execution of large numbers of database transactions by large numbers of people, typically over the internet. A database transaction is a change, insertion, deletion, or query of data in a database.

Step 7: Check assumptions: Check one machine learning model work as all condition region and country.

#Data Gathering

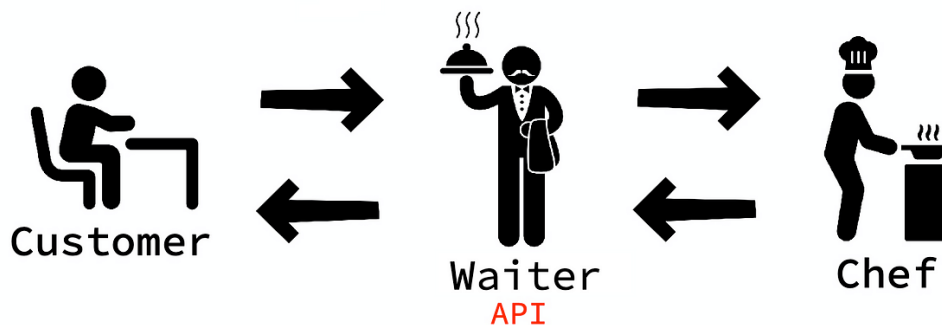
3. Fetching data from an API

What is the API and how it works?

API stands for “application programming interface.” An API is essentially a set of rules that dictate how two machines talk to each other. Some examples of API-based interactions include a cloud application communicating with a server, servers pinging each other, or applications interacting with an operating system.

API receives a request

Similar to how a waiter takes an order from a customer to relay to the chef



API collects and processes a response, then returns with that response

As a waiter would return the completed meal from the chef to the customer

Website we use for fetching API

TMDB: <https://www.themoviedb.org/>

TMDB API: <https://developer.themoviedb.org/reference/movie-top-rated-list>

Generate API Key after login and Paste in this link for JSON File

[https://api.themoviedb.org/3/movie/top_rated?api_key=<<api_key>>&lang
age=en-US&page=1](https://api.themoviedb.org/3/movie/top_rated?api_key=<<api_key>>&language=en-US&page=1)

Use tool for understand the JSON File: <https://jsonviewer.stack.hu/>

**Our objective: we want to create data set in python where row and column
8974 * 6**

ID | Title | Realised Date | Overview | Popularity | Vote average

Python Code

Day 4 Machine Learning | iScale

```
import pandas as pd
```

```
import requests
```

```
response =  
requests.get('https://api.themoviedb.org/3/movie/top_rated?api_key=5bc643f  
664c8c66e7edce11b77c7214e&language=en-US&page=1')
```

```
df =  
pd.DataFrame(response.json()['results'])[['id','title','release_date','overview','po  
pularity','vote_average']]
```

```
df.head()
```

```
response =  
requests.get('https://api.themoviedb.org/3/movie/top_rated?api_key=5bc643f  
664c8c66e7edce11b77c7214e&language=en-US&page=2')
```

```
df1 =  
pd.DataFrame(response.json()['results'])[['id','title','release_date','overview','po  
pularity','vote_average']]  
  
-----  
  
df1  
  
-----  
  
df1.shape  
  
-----  
  
df1.to_csv('movies.csv')
```

#Data Gathering

4. Fetching data using web scraping



Industries
Helping
Hands