ETL Project

Part 1: Reddit (HTML to MongoDB)

Reddit is a website where users post content that is then voted on by other users with the highest voted content appearing on the front page. Internet clickbait sites and lazy journalists will often copy and paste content from this site and call it news in an attempt to drive web traffic to their site. It's self-described as "the front page of the internet" and just like most sources of news, the headlines are worth reading, but the content not so much. In this project we automate the process of going to the site to read the headlines and put them into a MongoDB database to be read later.

Source

The data source for this project will be the Reddit homepage (https://www.reddit.com/) and five of its subreddits: r/science, r/technology, r/programming, r/AskReddit, and r/news. We will be using simple Python web requests to get the HTML, and BeautifulSoup to parse it.

Transformation

The site's content is organized in div elements. The CSS class names of the divs are not human readable, so we will instead be getting the headlines by finding the comments span element that common in each post and then using BeautifulSoup to find neighbor elements of interest. The data will be stored in a Python dictionary containing "title", "link", "age", "comments", and "subreddit" keys.

Destination

After the data has been scraped and placed into a Python dictionary using BeautifulSoup we will then store the dictionary as a document in a MongoDB collection. The documents will be inserted into a collection called "posts" in a database called "reddit". Note that MongoDB records a timestamp of all insertions and encodes them in the document's "$oid", thus eliminating the need to manually store when the data was scraped.

Example output from mongo shell of command: `db.getSiblingDB("reddit").posts.find()[17]`

```
{
  "_id": {"$oid": "5eadaeb3f6f02e88202f9891"},
  "title": "Did you know: Android was originally designed for digital cameras not
            phones",
  "link": "/r/technology/comments/gc595x/did_you_know_android_was_originally_
           designed_for/",
  "age": "5 hours",
  "comments": "21",
  "subreddit": "technology"
}
```

*Command to get timestamp: `db.getSiblingDB("reddit").posts.find()[17]._id.getTimestamp()`

```
ISODate("2020-05-02T17:32:35Z")
```

Part 2: Iris Flower Dataset (API to SQL)

The Iris flow dataset is a standard used to test statistical classification techniques. It contains a collection of sepal and petal measurements from three species of Iris flowers. For this assignment we will be playing the part of a student who wishes to practice their machine learning abilities, server side.

You have just implemented a k-means clustering SQL function (written for PostgreSQL in C) and you wish to test it. You know the Iris flower dataset would be perfect for this. You have used it often in the scikit-learn Python library, and figure it would be a sinch to write a small python program to output a SQL script version of the data.

Source

In this example the datasource is the scikit-learn Python library. It has several datasets it stores internally as CSVs, but it is best to read them using the library's APIs as the tables are not necessarily stored in an intuitive matter.

Transformation

Scikit-learn stores the Iris flower dataset in four different variables: feature_names, data, target_names, and target. First, these are combined into two variables: a table header, and a table body. Then, they are exported as a SQL script using low level Python string manipulation.

Destination

The output of the python script is a file called `iris.sql`. The head of the file is displayed below

```
DROP TABLE IF EXISTS IRIS;
CREATE TABLE IRIS (
  SEPAL_LENGTH FLOAT,
  SEPAL_WIDTH FLOAT,
  PETAL_LENGTH FLOAT,
  PETAL_WIDTH FLOAT,
  TARGET ENUM('setosa', 'versicolor', 'virginica')
);

INSERT INTO IRIS (SEPAL_LENGTH, SEPAL_WIDTH, PETAL_LENGTH, PETAL_WIDTH, TARGET)
  VALUES ('5.1', '3.5', '1.4', '0.2', 'setosa');
INSERT INTO IRIS (SEPAL_LENGTH, SEPAL_WIDTH, PETAL_LENGTH, PETAL_WIDTH, TARGET)
  VALUES ('4.9', '3.0', '1.4', '0.2', 'setosa');
INSERT INTO IRIS (SEPAL_LENGTH, SEPAL_WIDTH, PETAL_LENGTH, PETAL_WIDTH, TARGET)
  VALUES ('4.7', '3.2', '1.3', '0.2', 'setosa');
INSERT INTO IRIS (SEPAL_LENGTH, SEPAL_WIDTH, PETAL_LENGTH, PETAL_WIDTH, TARGET)
  VALUES ('4.6', '3.1', '1.5', '0.2', 'setosa');
INSERT INTO IRIS (SEPAL_LENGTH, SEPAL_WIDTH, PETAL_LENGTH, PETAL_WIDTH, TARGET)
  VALUES ('5.0', '3.6', '1.4', '0.2', 'setosa');
```

The table can be loaded using the command '`mysql -u username -p database_name < iris.sql`' and the table can be queried using the SQL statement '`SELECT * FROM IRIS LIMIT 5`'

SEPAL_LENGTH	SEPAL_WIDTH	PETAL_LENGTH	PETAL_WIDTH	TARGET
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa