Zach Shim

Capstone Proposal

CSS 490

**Description:**

Fake reviewers promote false advertisement, which negatively affects the business of sellers and the experience of customers.

**System Architecture**

Model-View-Controller (MVC) architectural pattern.

**View**:

User picks a product to detect fake reviews for.

**Controller**:

Uses machine learning algorithms to determine validity of reviews for a product.

The system architecture is broken into three lowly coupled parts:

1. Detection of duplicate reviews
2. Detection of anomaly in review count and rating distribution
3. Detection of incentivized reviews

**Model**:

A credibility score is generated and displayed to the user.

**Requirements:**

* Improve code readability and documentation.
* Importing CSV/JSON files to a database
  + The entirety of the program parses data files directly. I feel like it would be much simpler if the base data in csv/json files were first imported into a SQLite database which can be kept on the server, and any additional data that needs to be handled can be create a temporary (by temporary I mean the database is kept in ram and will disappear once execution is halted). This will stop wait times from having to continuously load a bunch of data files and allow for simpler parsing of data.
* Deploy the shell Python app to a website - hosted on GitHub or UW student site
  + May slightly change the UI, but mostly keep it true to the original due to my lack of experience with web development and I’m hoping this is one of the things this project will allow me to explore
* Allow user to input a review
  + Allows user to see real-time data instead of relying on a static database.
  + Scraping directly from an amazon link given a ASIN key
* Add another machine learning methodology
  + Possible implementation: logical regression or Linear SVM

**Tech Stack**

* Backend:
  + Django
  + Python
  + SQLite3
* Frontend:
  + HTML5
  + CSS
  + Javascript

**What I Hope to Learn:**

* Experience with full stack development
  + Web development
  + Database manipulation in python
* Machine learning and data mining concepts and algorithms

**Scaling:**

* **Website**
  + Extending this into a web-based service hosted on GitHub.
  + Utilizing MySQL to keep data up to date.
* **Security**
  + Filter data (input validation controls)
    - A certain data element not allowed in a query
    - Not going to allow code, looking at something that is just text
  + Browser defenses
    - Automatic filtering, checks for malicious code
  + Consider input as text (innerHTML vs text)
    - Not going to allow code, looking at something that is just text
    - Do not execute it, classify it as text. Even if a script is put in, it is just text
    - innerHTML – accept any input and assume at times that HTML code will allow me to execute another query. Considered HTML within HTML. Can do a query within a query to get additional information.

**Application:**

Any small or large business that displays reviews on their website.

Plagiarism detection.

**Future Implications:**

When pulling data from the dataset, eval() interprets a string as code. Therefore, if a malicious user were to insert code into the database, they can run harmful code. If you have eval(input()) and os imported, a person could type into input() os.system('rm -R \*') which would delete all your files in your home directory. (Assuming you have a unix system).