Phase 1: Text Search

I am doing a Movie Search Engine. So, on the server, a user can enter few keywords and from that keywords, we would rank what all movie are relevant to that free text.

How do we do that:

- Preprocess our csv file.
- Tokenize the plot, converting our plot to the lower case, remove stop words.
- Create an inverted index.
- Build our document vector in which we calculate our TF-IDF scores.
- After that we calculate cosine similarity.
- After that save the inverted index in a .pkl file which will improve our searching speed.
- We then process our query or keywords that the user had entered and calculate total tf-idf score, idf score for individual query term and tf score for individual query term.
- At last we print title, plot, tf-idf score, idf score, tf score and IMDB rating.
- Deploying using Pythonanywhere is the last step.

Dataset: https://www.kaggle.com/beyjin/movies-1990-to-2017#Movie_Movies.csv

Highlights of this project:

- Stemming VS Lemmatization
- Total Tf-IDF score
- TF and IDF for individual query terms

TF-IDF Total Score Equation:

$$w_{t,q} = (1 + log_{10} tf_{t,q}).$$

TF-IDF Equation:

$$w_{t,d} = (1 + log_{10} tf_{t,d}) \times (log_{10} \frac{N}{df_t}).$$

Keyword: Kid alone at home

Without tf-idf normalization:

(**'Home Alone: Purged'**, "There's something of value in an old family home. A group of scavengers gather to take it. Left home alone, 3 kids must try to stop them.", 3.800885136649116, **nan**),

(**'Kids vs. Zombies'**, 'A young brother and sister work together to outwit a barrage of peculiar zombies, rescue their mom and save the town. Action-packed, gut-busting zombie fun for all ages. "Home Alone" meets "Zombieland".', 3.110899569636624, **nan**),

(**'The Gate'**, 'Kids, left home alone, accidentally unleashes a horde of malevolent, pint-sized demons from a mysterious hole in their suburban backyard.', 3.110899569636624, **6.0**),

(**'Home Alone Horror'**, 'A boy left home alone realizes he may not be totally alone.', 2.7523199410205397, **6.6**),

(**'Alone'**, "Erin is spending her first night home alone in many years. Even though she's locked safely inside, every bump, every creak, every little sound convinces her she is not alone.", 2.5690183438974192, **nan**)

With tf-idf normalization:

('A Girl Walks Home Alone at Night', 'A girl walks home alone at night and a man starts to follow her.', 0.45904343187331254, **6.5**),

(**'Alone'**, 'A man who is stuck alone in his home and is being haunted by his dead wife.', 0.4571230787656072, **nan**),

(**'Home'**, 'Home is where you are.', 0.4424205006970245, nan),

(**'Home Alone Horror',** 'A boy left home alone realizes he may not be totally alone.', 0.4412866356710031, **6.6**)

(**'Home Alone: Purged'**, "There's something of value in an old family home. A group of scavengers gather to take it. Left home alone, 3 kids must try to stop them.", 0.4083518533240615, **nan**)

Lemmatization:

(**'Alone'**, 'A man who is stuck alone in his home and is being haunted by his dead wife.', 0.4432479421067452, **nan**),

(**'Home'**, 'Home is where you are.', 0.4358973946078705, **nan**),

(**'Home Alone Horror'**, 'A boy left home alone realizes he may not be totally alone.', 0.4139130726129726, **6.6**),

(**'Home Alone: Purged'**, "There's something of value in an old family home. A group of scavengers gather to take it. Left home alone, 3 kids must try to stop them.", 0.39285220213737304, **nan**),

(**'A Girl Walks Home Alone at Night'**, 'A girl walks home alone at night and a man starts to follow her.', 0.3754790320456582, **6.5**)

References:

1. https://nlp.stanford.edu/IR-book/pdf/13bayes.pdf
2. https://gist.github.com/adolfoguimaraes/91fbef8beceabafdcef2b407639290d4
3. https://www.geeksforgeeks.org/python-lemmatization-with-nltk/