What is Recommendation system?

Presentation by Team Solo

Recommendation system

- During the last few decades, with the rise of Youtube, Amazon, Netflix and many other such web services, recommender systems have taken more and more place in our lives.From e-commerce to online advertisement, recommender systems are today unavoidable in our daily online journeys.
- Now the main purpose of the recommender system is to suggest relevant items to the user based on various parameters.
- Recommendation algorithms can be divided into two great paradigms:collaborative approaches (such as user-user, item-item and matrix factorisation)that are only based on user-item interaction matrix and content based approaches (such as regression or classification models) that use prior information about users and/or items.

Problem

How can you predict what a user wants to buy based on the content user consumes?

- ➔ To predict as precisely as possible we might want to know what users consume as content. Now content varies depending upon individuals liking for eg:- blogging, videos, images, etc. And we also know that in this internet era Image or videos are the Primary sources of sharing data.

- ➔ Now keeping that in mind I choose an image-based approach to solve this problem. Because Image is the most shared type of data and we also know that every product has an image attached to it for a better understanding and to create trust of the consumers.

Why Image Based Recommendation System?

- Recommendation system has always been in demand in every industry and domain as matching a user's preferences is always the most important thing in modern-day business.

- The Project deals with recommending the most suitable products to Users based on the Current Selection and choice of the product. Our model relies on the fact that multiple features can be extracted from images and used for similarity computation.

- Now the Core Idea is to use State of the art Deep learning Techniques to Extract features from the image and convert those images into vectors and use similarity computation to recommend similar products.

Data Collection

- There are Lot of Ways to Collect Data. Since our Need is image-based Data we can use web scraping techniques or we can use some APIs to import the data and kaggle has around 25gb worth of Data images of various products. That can be used for training and test purposes. Thanks to kaggle.

- But in this Project I am using a Sample data Obtained from Amazon Product Advertising Api. To train the model and to get the desired result.

- Now the Data that i have obtained has 183138 number of data points and 19 features. Now amongst all the feature there is one feature called large image URL using this feature we gonna download all the images and use it to train the model. The Dataset largely Contains Women's top.

Data Processing

- Data preprocessing involves the transformation of the raw dataset into an understandable format. Preprocessing data is a fundamental stage in data mining to improve data efficiency. The data preprocessing methods directly affect the outcomes of any analytic algorithm.
- Steps In Data Preprocessing:
  - Gathering all the data
  - Importing all the dataset and libraries
  - Dealing with all the missing values
  - Removing duplicates
  - Removing similar data
  - Loading all the necessary file in pickle
  - Removing stopwords
  - And Some Text Processing

Text based product similarity

- Now as our base model I have used a text-based Product similarity. In This, I have used Bag of words and Term Frequency Inverse Document Frequency(TFIDF) on the title feature which creates a set of a corpus, and now using euclidean distance we are recommending similar products.
- What is Bag of words?
  - A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms.
  - A bag-of-words is a representation of text that describes the occurrence of words within a document.
  - It basically Counts How many times words occur in the dataset and makes the Set of dictionary.
  - After Getting the set of count of unique words on the Title features and then using Euclidean Distance on the title feature to get the similar products.

➔ What is Term Frequency - Inverse Document Frequency(TFIDF)?

- **tf-idf** stands for *Term frequency-inverse document frequency*. The tf-idf weight is a weight often used in information retrieval and text mining. Variations of the tf-idf weighting scheme are often used by search engines in scoring and ranking a document's relevance given a query.
- This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus (data-set).
- TF-IDF for a word in a document is calculated by multiplying two different metrics:
  - The **term frequency** of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by length of a document, or by the raw frequency of the most frequent word in a document.
  - **The inverse document frequency** of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.
  - So,if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1.
  - Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document

➔ After Getting the Tfidf vectors on the Title features and then using Euclidean Distance on the title feature to get similar products.

Deep Learning Based approach

- The results achieved with Text based similarity were not convincing enough for us and therefore, we decided to explore techniques which may help us extract more descriptive and distinctive feature representation for the images which can be further used in computing similarity score based on which top K recommendations can be returned to the user. In a quest to find such feature representation, we landed at Deep Learning techniques which are quite powerful at extracting patterns and features from images.
- Pre-trained models based features extraction technique:
  - For this purpose, we mainly used one of the most popular CNN based architectures namely VGG16 in which VGG stands for Visual Geometry Group and 16 basically means Its a 16 layer neural network.
  - To use pre-trained deep learning models as feature extractors, the very first step was to remove its final output layer as we did not intend to use these models as classifiers.
- What Vgg16 basically does is it takes image as an input and outputs is at d-dim dense vector.
- So after passing almost 16k images to VGG16 it converts each image into 25088 length dense vector. I ran it for 30 epochs, batch-size was 1.

Visual Feature Based Product Similarity

- After getting 25088 length dense vector i am saving it in a file called 16k_data_cnn_features.npy
- After getting all the features similarly and sorting all the features and computing the euclidean distance to get similar products.
- Now weights i have taken from the imagenet dataset and trained our model on those weights.
- When I gave the input to this model and i asked for 10 similar products it gave us a pretty good result as compared to the text-based similarity model.
- Now the Entire model was trained on the Images and result produced are from the image as well.

What is Next?

- ➔ We Can use more Pre-trained models like Resnet50 , Inception, etc to train our models and compare the result of the models.
- ➔ We Can build or own CNN and train our model using that CNN.
- ➔ We Can Also perform Hyperparameter tuning and rewrite the entire top layer.
- ➔ We can also use dropout,dense,flatten methods to increase the overall efficiency of the model.
- ➔ We can also use a Generative adversarial neural network or GANS in short for getting accurate recommendations.
- ➔ We can also Combine the top working model or Use a hybrid model for recommending products and we can also use Data Augmentation for better efficiency.