**AMAZON RECOMMENDATION ENGINE**

**1.1-Introduction:**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), artificial intelligence (AI), sometimes called machine intelligence, is [intelligence](https://en.wikipedia.org/wiki/Intelligence) demonstrated by [machines](https://en.wikipedia.org/wiki/Machine), in contrast to the natural intelligence displayed by humans. Colloquially, the term "artificial intelligence" is used to describe machines/computers that mimic "cognitive" functions that humans associate with other [human minds](https://en.wikipedia.org/wiki/Human_mind), such as "learning" and "problem solving".

Artificial intelligence can be classified into three different types of systems:

* Analytical
* Human-inspired
* Humanized artificial intelligence.

Analytical AI has only characteristics consistent with [cognitive intelligence](https://en.wikipedia.org/wiki/Cognition); generating cognitive representation of the world and using learning based on past experience to inform future decisions. Human-inspired AI has elements from cognitive and [emotional intelligence](https://en.wikipedia.org/wiki/Emotional_intelligence); understanding human emotions, in addition to cognitive elements, and considering them in their [decision making](https://en.wikipedia.org/wiki/Decision_making). Humanized AI shows characteristics of all types of competencies (i.e., cognitive, emotional, and [social intelligence](https://en.wikipedia.org/wiki/Social_intelligence)), is able to be [self-conscious](https://en.wikipedia.org/wiki/Self-consciousness) and is [self-aware](https://en.wikipedia.org/wiki/Self-awareness) in interactions with others.

**Need for Artificial Intelligence**

* To create expert systems which exhibit intelligent behavior with the capability to learn, demonstrate, explain and advice its users.
* Helping machines find solutions to complex problems like humans do and applying them as algorithms in a computer-friendly manner.

**Applications of artificial intelligence**

* Knowledge reasoning.
* Planning.
* Machine learning.
* Natural language processing.
* Computer vision.
* Robotics.
* Artificial general intelligence.

Artificial intelligence is considered to be the trending technology of the future. Already there are a number of applications made on it. Due to this, many companies and researchers are taking interest in it. But the main question that arises here is that in which programming language can these AI applications be developed? There are various programming languages like Lisp, Prolog, C++, Java and Python, which can be used for developing applications of AI. Among them, Python programming language gains a huge popularity and the reasons are as follows −

**Simple syntax & less coding**

* Python involves very less coding and simple syntax among other programming languages which can be used for developing AI applications. Due to this feature, the testing can be easier and we can focus more on programming.

**Inbuilt libraries for AI projects**

* A major advantage for using Python for AI is that it comes with inbuilt libraries. Python has libraries for almost all kinds of AI projects. For example, NumPy, SciPy, matplotlib, nltk, SimpleAI are some the important inbuilt libraries of Python.
* **Open source** − Python is an open source programming language. This makes it widely popular in the community.
* **Can be used for broad range of programming** − Python can be used for a broad range of programming tasks like small shell script to enterprise web applications. This is another reason Python is suitable for AI projects.

**Features of Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python's features include the following −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode**− Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − We can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases**− Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

**Important features of Python**

Let us now consider the following important features of Python −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**1.2 Objectives of Research**

Online shopping is all over the internet. All our needs are just a click away. The biggest online shopping website is Amazon. Amazon is known not only for its variety of products but also for its strong recommendation system.

In our project we are taking into consideration the amazon review dataset for Clothes, shoes and jewelleries and Beauty products. We are considering the reviews and ratings given by the user to different products as well as his/her reviews about his/her experience with the product(s).

Based on these input factors, sentiment analysis is performed on predicting the helpfulness of the reviews. Moreover, we also designed item-based collaborative filtering model based on k-Nearest Neighbors to find the 2 most similar items.

**1.3 Problem Statement**

The most daunting aspect of building a recommendation engine is knowing where to start. This is even more difficult when you have limited or little experience with ML. However, you may be lucky enough to know what you don’t know, such as:

* What data to use.
* How to structure it.
* What framework/recipe is needed.
* How to train it with data.
* How to know if it’s accurate.
* How to use it within a real-time application.

**2. Review of literature**

Recommendation System belongs to the class of Information Retrieval, Data Mining and Machine Learning. Recommender systems play a major role in today's ecommerce industry. Recommender systems recommend items to users such as books, movies, videos, electronic products and many other products in general. Recommender systems help the users to get personalized recommendations, helps users to take correct decisions in their online transactions, increase sales and redefine the users web browsing experience, retain the customers, enhance their shopping experience. Information overload problem is solved by search engines, but they do not provide personalization of data. Recommendation engines provide personalization. There are different type of recommender systems such as content-based, collaborative filtering, hybrid recommender system, demographic and keyword based recommender system. Variety of algorithms are used by various researchers in each type of recommendation system. Lot of work has been done on this topic, still it is a very favourite topic among data scientists. It also comes under the domain of data Science.

**3. Data Collection**

**Sources:** <https://www.kaggle.com/qwikfix/amazon-recommendation-dataset>

This dataset consists of reviews of various products from amazon. The data span a period of more than 10 years, including all ~50,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

The purpose of this analysis is to make up a prediction model where we will be able to predict whether a recommendation is positive or negative. In this analysis, we will not focus on the Score, but only the positive/negative sentiment of the recommendation.

To do so, we will work on Amazon's recommendation dataset, we will build a Term-doc incidence matrix using term frequency and inverse document frequency ponderation. When the data is ready, we will load it into predicitve algorithms, mainly naïve Bayesian and regression.

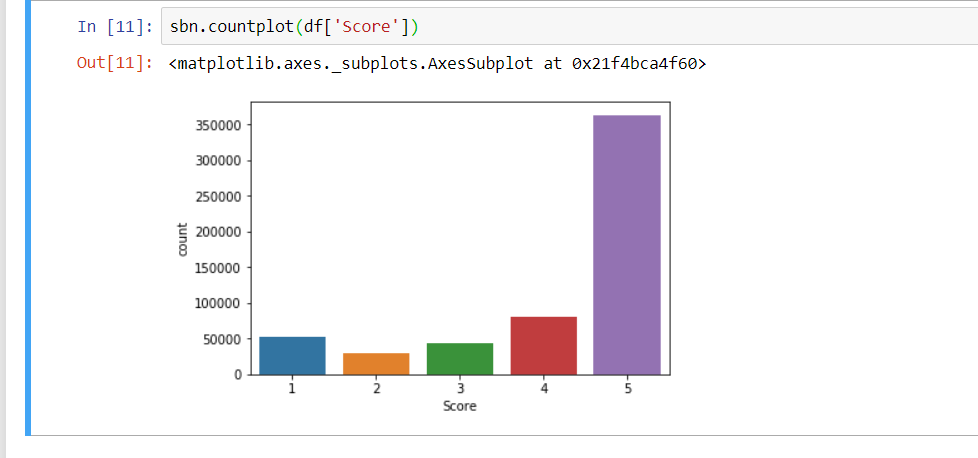
In the end, we hope to find a "best" model for predicting the recommendation's sentiment.

We use the first dataset to compute this score. For a target product, we get all the reviews for that product and for each review we extract the helpfulness vote and the overall score. The thought process is as follow: the higher the overall score the better is the product. Also the higher the helpfulness the more people considered this review when buying this product. So we need to build a product score that is directly proportionate to the overall score and the helpfulness vote

**4. Methodology**

**4.1 Exploratory Data Analysis:**

**4.1.1 Figure’s and Tables:**

****

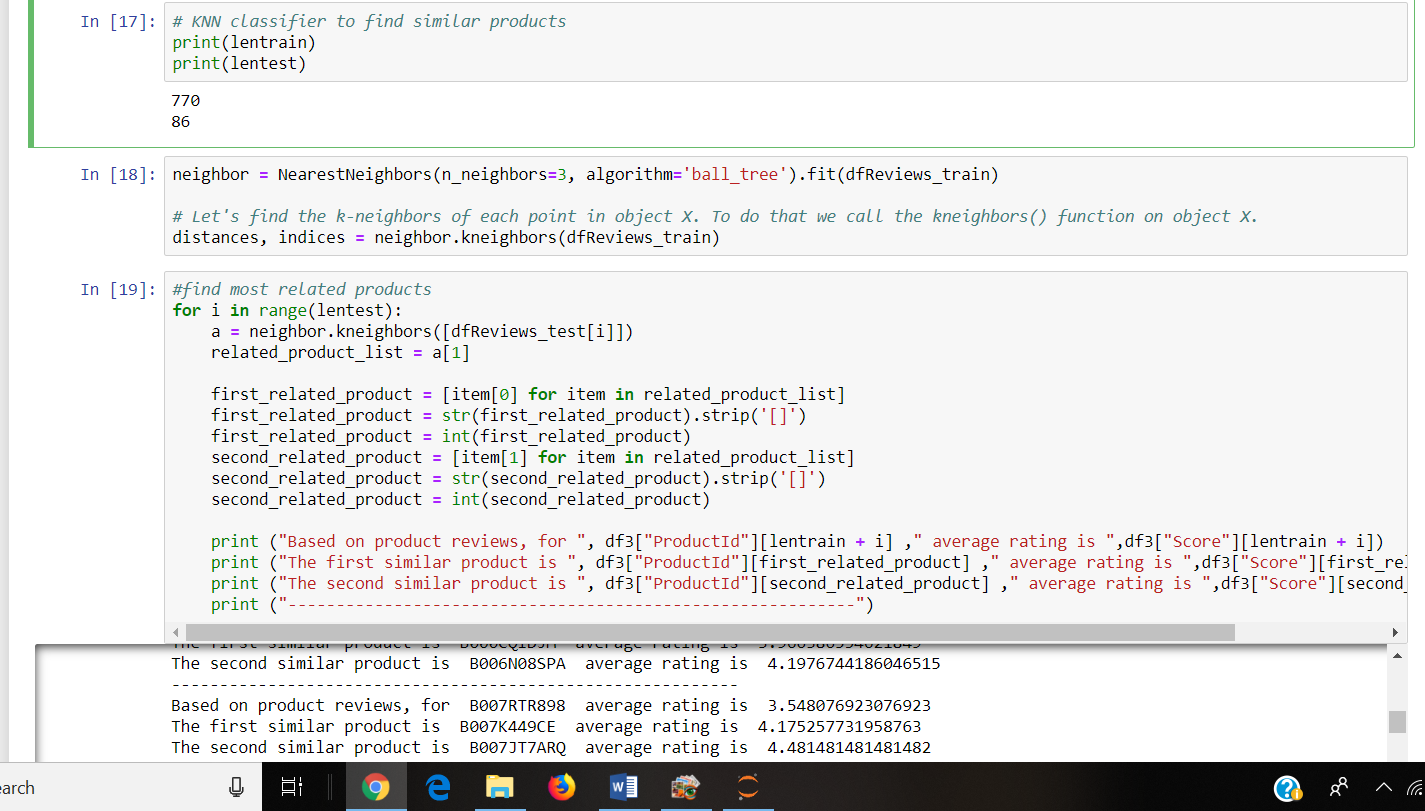
**4.2 Data Modelling:**

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as [GMM](https://en.wikipedia.org/wiki/Mixture_model), which assume a Gaussian distribution of the given data).

In k-NN regression, the k-NN algorithm is used for estimating continuous variables. One such algorithm uses a weighted average of the k nearest neighbours, weighted by the inverse of their distance. This algorithm works as follows:

* Compute the Euclidean or [Mahalanobis distance](https://en.wikipedia.org/wiki/Mahalanobis_distance) from the query example to the labeled examples.
* Order the labeled examples by increasing distance.
* Find a heuristically optimal number k of nearest neighbors, based on [RMSE](https://en.wikipedia.org/wiki/RMSE). This is done using cross validation.
* Calculate an inverse distance weighted average with the k-nearest multivariate neighbors.



**5. Findings and Suggestions**

**Sources:** <https://www.kaggle.com/qwikfix/amazon-recommendation-dataset>

<https://github.com/mandeep147/Amazon-Product-Recommender-System/blob/master/Recommender%20System/Recommender%20System.ipynb>

Given more time to work on the project, we had a couple of ideas to further implement. One of them is to preform NLP on the review text to extract sentiment from each review. We would attribute a score to how positive or negative reviews is predicted. This would lead us to compute a different product score function. We can further boost the product score by using the rank of the product in its category given in the dataset.

Evaluation wise, we can add more products and more participants to get better estimation of the relevancy of the products. This would be streamline by creating an app that proposes recommendations for users.

**6. Conclusion**

Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. For large retailers like Amazon.com, a good recommendation algorithm is scalable over very large customer bases and product catalogs, requires only subsecond processing time to generate online recommendations, is able to react immediately to changes in a user’s data, and makes compelling recommendations for all users regardless of the

number of purchases and ratings. Unlike other algorithms, item-to-item collaborative filtering is able to meet this challenge. In the future, we expect the retail industry to more broadly apply recommendation algorithms for targeted marketing, both online and offline. While e-commerce businesses have the easiest vehicles for personalization, the technology’s increased conversion rates as compared with traditional broad-scale approaches will also make it compelling to offline retailers for use in postal mailings, coupons, and other forms of customer communication.