# **Statement of Work**

# Develop Smart Recommendation System in Ecommerce

**Course Code/Name**

AIDI-1002-02 AI ALGORITHMS I

**Course Facilitator**

Marcos Bittencourt

**Student Name**

Ayush Parolia

**Student Id**

100799278

Date: - November 05, 2020

# **Executive Summary**

To start with lets with a basic question - What is Recommendation System?

A recommendation system often termed as Recommender Systems, they are simple algorithms which aim to provide the most relevant and accurate items to the user by filtering useful stuff from of a huge pool of information base.

Recommendation engines discovers data patterns in the data set by learning choices and produces the outcomes that co-relates to their needs and interests.

Real time Examples: - Amazon and Netflix using recommendation engine for suggesting what customer might also like and is as per their interest.

**Scope**

We are asked to create a system that automatically recommends a certain number of products to the consumers on an E-commerce website based on the past purchase behavior of the consumers.

* Word2vec is an ultra-popular word embeddings used for performing a variety of [NLP](https://courses.analyticsvidhya.com/courses/natural-language-processing-nlp?utm_source=blog&utm_medium=how-to-build-recommendation-system-word2vec-python) tasks
* We will use word2vec to build our recommendation system.

**Goals and Objectives**

* Create customer-centric search
* Retarget potential customers
* Identify exceptional target prospects
* Create a more efficient sales process
* Improve recommendations for customers

# **Analytics Rationale Statement**

The dynamic sector that is eCommerce, has revolutionised the way a consumer shops in our mobile world. The desire of many eCommerce businesses is to bring the best of an offline shopping experience to the online space, by offering customers a seamless way to discover products they are actively looking for.

There is an important focus in ‘hyper personalisation’, which could only be facilitated by learning genuine consumer behaviour and making predictions with gargantuan amounts of data that is collected from user activities on smartphones, tablets and desktops.

Traditionally, we would like to buy a product that friends or colleagues has suggested. In a digital world using these kinds of strategies as recommendation systems, the product owner can recommend items that customers might also liked and required.

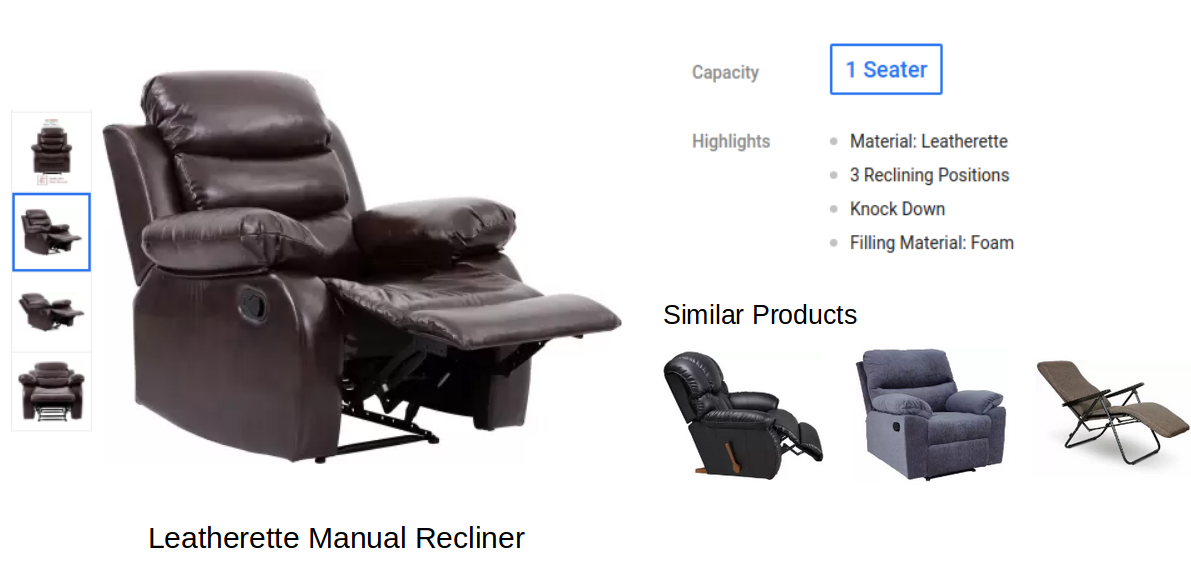
How does YouTube know what videos you’ll watch? How does Google always seem to know what news you’ll read? They use a Machine Learning technique called Recommender Systems. Practically, recommender systems encompass a class of techniques and algorithms which are able to suggest “relevant” items to users. Ideally, the suggested items are as relevant to the user as possible, so that the user can engage with those items: YouTube videos, news articles, online products etc.

Items are ranked according to their relevancy, and the most relevant ones are shown to the user. The relevancy is something that the recommender system must determine and is mainly based on historical data. If you’ve recently watched YouTube videos about elephants, then YouTube is going to start showing you a lot of elephant videos with similar titles and themes!

# **Problem Statement**

Consumers often abandon eCommerce experiences because the product results displayed are often irrelevant and hence an organisation or company loses its potential customer.

Let me use a recent example to showcase their power. I went to a popular online marketplace looking for a recliner. There were hundreds of them. From Traditional two-position recliners to Push-Back Liners; from Power Lift Recliner to the Wall Hugger one. I liked most of them and I clicked on a leatherette manual recliner:



Notice the different kinds of information presented on this page. The left half of the image contains the pictures of the product from different angles. The right half contains a few details about the product and a section of similar products.

This is my favourite part of the image. The website is recommending me similar products and it saves me the effort to manually go and browse similar armchairs.

# **Data Requirements**

For the analysis, the data required will be fetched from the online platform of E-commerce for e.g. Amazon or Flipkart.

Here is the description of the fields in this dataset:

InvoiceNo: Invoice number. a unique number assigned to each transaction

StockCode: Product/item code. a unique number assigned to each distinct product

Description: Product description

Quantity: The quantities of each product per transaction

InvoiceDate: Invoice Date and time. The day and time when each transaction was generated

CustomerID: Customer number. a unique number assigned to each customer

We have identified two constraints with the data:

* The dataset should not have empty or blank columns for purchase history

And assumptions about the dataset:

* That the data is valid and for an individual.

# **Model Approach**

## **Introduction to word2vec – Vector Representation of Words**

We know that machines struggle to deal with raw text data. In fact, it’s almost impossible for machines to deal with anything except for numerical data. So, representing text in the form of vectors has always been the most important step in almost all NLP tasks.

One of the most significant steps in this direction has been the use of word2vec embeddings, introduced to the NLP community in 2013. It completely changed the entire landscape of NLP. These embeddings proved to be state-of-the-art for tasks like word analogies and word similarities. word2vec embeddings were also able to achieve tasks like King – man +woman ~= Queen, which was considered an almost magical result. Now, there are two variants of a word2vec model ⁠— Continuous Bag of Words and Skip-Gram model. In our project, I will use the Skip-Gram model. Let’s first understand how word2vec vectors or embeddings are calculated.

**How are word2vec Embeddings Obtained?**

A word2vec model is a simple [neural network](https://www.analyticsvidhya.com/blog/2017/05/neural-network-from-scratch-in-python-and-r/?utm_source=blog&utm_medium=how-to-build-recommendation-system-word2vec-python) model with a single hidden layer. The task of this model is to predict the nearby words for each and every word in a sentence. However, our objective has nothing to with this task. All we want are the weights learned by the hidden layer of the model once the model is trained. These weights can then be used as the word embeddings.

Let me give you an example to understand how a word2vec model works. Consider the sentence below:[word2vec example](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/07/img_1.png)

Let’s say the word “teleport” (highlighted in yellow) is our input word. It has a context window of size 2. This means we are considering only the 2 adjacent words on either side of the input word as the nearby words.

Note: The size of the context window is not fixed, it can be changed as per our requirement.

Now, the task is to pick the nearby words (words in the context window) one-by-one and find the probability of every word in the vocabulary of being the selected nearby word.

