Creative Computing Year 4

Artificial Intelligence CA 1

Building a Content based recommender system.

Gary Thompson (N00161709)

Contents

1. Introduction.
2. Applications of AI
3. Recommender Systems
4. Code
5. Conclusion
6. References
7. Introduction

Artificial intelligence is the concept of training a computer to simulate human intelligence. Artificial Intelligence, or AI as its more commonly dubbed, uses various algorithms for different tasks in order to make independent decisions without human interference. There are 4 distinct types of AI, reactive, limited memory, theory of mind, and self-aware. In this report, the various applications and methods of AI will be discussed, as well as a specific use case for AI known as a recommender system. Moreover, a recommender system that was designed in tandem with this report will be explored thoroughly. Recommender systems are seen frequently in the modern world, with sites such as Netflix, Steam, Amazon, and Spotify recommending media or products based on what you have shown interest in prior. There are different styles of recommender system such as Popularity based, Collaborative based, and Content based. A comparative will be drawn between these types, and a Content based recommender will be explored in detail. The recommender system was designed with Jupyter Notebook and the Python language. It utilizes a mobile game store dataset to make similar recommendations to a chosen game that is entered.

1. Applications of AI

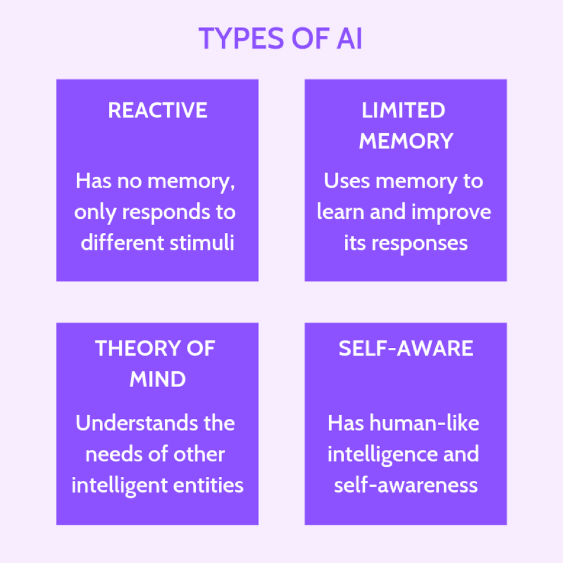
To understand the diverse range of AI applications we must first look at AI from its conception and how its evolved in the modern day. AI has been around for about 60 years, its first iteration manifesting from great minds like Donald Michie, Richard Gregory and Christopher Longuet-Higgins, who believed that emulating the human cognitive process in computers could create either intelligent robots, or perhaps give a new theoretical perspective on nature. AI now represents computers abilities to use human like decision making to figure out complex problems such as computation, chess, decision making, and recommendations. This is done with complex algorithms that are developed for the sole purpose of delivering a result for a specific task at hand, allowing computers to recognize patterns or dynamically derive meaning from data that’s presented. AI is a technology that is transforming every walk of life. It is a wide-ranging tool that enables people to rethink how we integrate information, analyze data, and use the resulting insights to improve decision making (West, Allen, 2018).

However, AI also poses a unique threat in the sense that as it can self-learn, what is to stop it from teaching itself skills or agendas that are not in the human race’s best interests? “The real problem relates to the possibility that AI may become incredibly good at achieving something other than what we really want” (Russel, 2016). This is a fear that is shared between the average person and computer scientists alike, due in part to Hollywood productions such as iRobot, the matrix and the terminator instilling fear in us, but also the fact that our scope of understanding towards artificial intelligence is relatively low. Even if AI doesn’t take such a hostile approach, there are fears of jobs becoming increasingly difficult to obtain due to the growth of use cases for AI. Whilst this is a widespread fear, not very many people consider the ripple effect of automation on the job market. The evidence shows that automation reduces costs and frees up labour, which allows further economic growth and new jobs in areas of demand that were unexpected (Halal et al, 2016).

According to a blog post by Naveen Joshi of Forbes, there are 7 core types of AI.

These are:

1. Reactive
2. Limited Memory
3. Theory of Mind
4. Self-Aware
5. ANI
6. AGI
7. ASI



Reactive AI programs were the first AI’s. They display the ability to emulate human like reactions to external stimuli, and they do not remember data or hold any information after a calculation is complete so they cannot learn from past actions. The most notable reactive AI was IBM’s deep blue which in 1997, beat Chess Grandmaster Garry Kasparov.

Limited Memory AI are like Reactive AI’s however they can remember data and previous experiences in order to inform their future decisions. When a deep learning model is being trained, it is more often than not a Limited memory AI, using vast amounts of data to form a point of reference for when it is asked to carry out a task. An example of a limited memory AI is a self-driving car such as the Tesla model S, which uses data based on maps, images of stoplights and road signs, and human driving behaviour to effectively emulate a person who is in the eyes of the law a good driver.

Theory of Mind is less similar to Reactive and limited memory and is currently only conceptual. This AI concept is currently in development across the world, with its sole purpose being to understand human emotions, needs, unpredictability and make choices, suggestions, and actions that benefit humankind. This is difficult to develop as whilst AIs are complicated so are Humans, and there is a lot of unpredictability and variables in order to get this effectively off the ground.

Self-Aware AI is currently only theoretical. This is the final goal of AI but also the most controversial. It basically represents a computer’s ability to be a sentient being of high intellect like a human. Not only will it be able to act and develop reactions to human emotions like theory of mind, but it will also have the ability to feel emotion, express it, and have opinions on things such as politics, the world, and what most fear, self-preservation. If we train AIs to act and think like humans, whose primary function is survival and the growth of the human race, what is to stop an AI from valuing this too much to the point where it sees humanity as a threat. This technology is decades away at the earliest but proper planning and thought must go into this before its developed to ensure humanity’s best interests are met.

ANI or Artificial narrow intelligence is an umbrella term for every Ai that has been conceived so far. It refers to any AI that can carry out a task with Human like actions and decision making. They are only capable of operating within the constraints that humans have put on them and are all within Reactive and limited memory classification.

AGI or Artificial general intelligence describes an AI that contains an Agent or Agents that are capable of functioning like an intellectual human being. In order to understand AGI, we first need to understand Agents, which are things within AI that Act, autonomously, recognise patterns in their environments, be adaptable, and be self-driven in terms of its goals and wants.

ASI or Artificial Superintelligence is the classification of AI that will be the point at which AI gains sentience. Not only that but due to complex programming, processing, and a massive amount of available data, they will be better than us at just about everything. This is an interesting but terrifying prospect but is an incredibly long time away.

There are several incredible applications of modern ai that we use every day. This report will discuss Computer vision, Machine learning, Deep learning, Natural Language processing and robotics.

Computer vision

A computer vision system processes images acquired from an electronic camera, which is like the human vision system where the brain processes images derived from the eyes (Nixon et al,2002). Computer vision is a subset of Artificial intelligence that an app such as is being discussed would use as its primary function. Computer vision historically has found many practical uses, specifically in manufacturing and production to find flaws in the production line, but also has ties in radiology, microscopy, and document processing. It allows companies to save large quantities of capital due to increased efficiency in many areas and paves the way for new research previously unfounded. Computer vision is used in industries ranging from energy and utilities to manufacturing and automotive – and the market is continuing to grow. It is expected to reach USD 48.6 billion by 2022 (IBM, 2021).

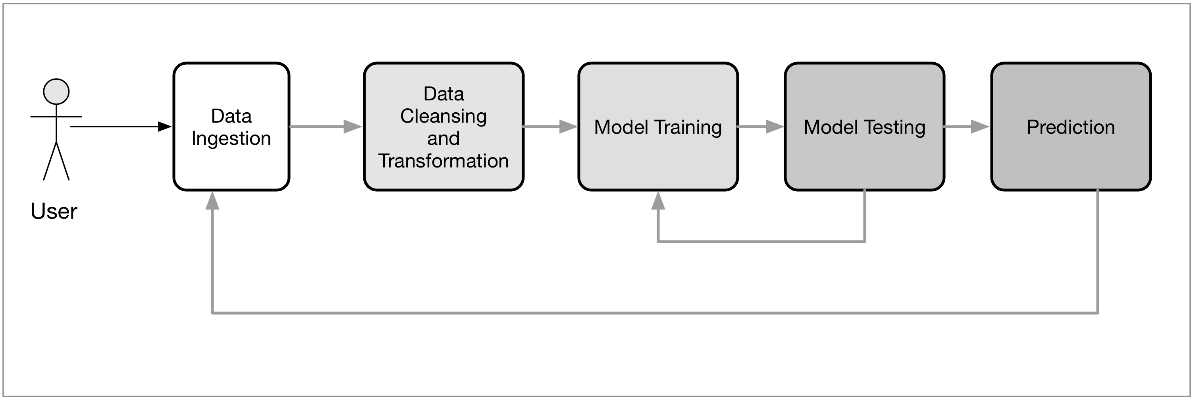
Important subsects of Computer vision are Image processing and Image recognition.

These are two sides of the same coin, and often are used in tandem with each other. Image processing is where a digital image is altered in some way such as colourblind filtering, image restoration, colour processing and microscopic imaging. This allows for image processing to be incredibly relevant technology in such industries as historical image restoration, medicine, photography, accessibility and advertising.

Image recognition on the other hand is more specifically centred around training a computer to be able to identify different properties in an image such as colour or patterns. This has many applications such as Facial recognition, Colour recognition, Object Recognition, Augmented reality, Vehicle autonomy and more. Common industries that see a lot of image recognition use are Medicine, Law enforcement, Robotics, Manufacturing, and quality control.

Machine Learning

Machine learning is a subset of AI that uses a vast number of algorithms to allow a program that has access to data, to use that data to learn and inform future decisions and allow for new conclusions to be drawn. There are 3 primary subsects of machine learning, Computational learning theory, Grammar induction and meta learning.



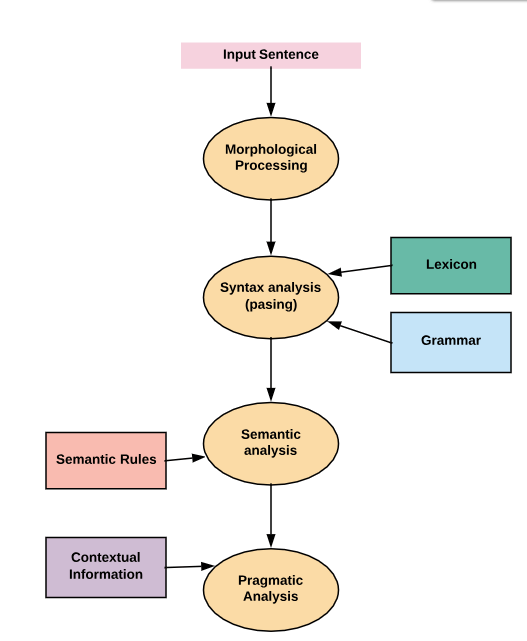
Computational learning theory is a subsect of Machine learning that explores the use of mathematical formulas applied to learning systems in order to apply computer science techniques to more abstract issues. One can extend statistical learning theory by taking computational complexity of the learner into account. This field is called computational learning theory or COLT. (Murphy, 2012). Computational learning theory applications can be seen in the likes of probability, statistics, programming optimization, calculus, and geometry.

Grammar induction is a subset of machine learning that according to the likes of Klein and Kuppin, is a way for computers to formal grammar and map them to meanings. This is known as the grammatical mapping paradigm. Aspects of this have been transferred to another subset of AI, Natural language processing, however it can also be found in translation, grammar-based compression, and semantic parsing.

Meta learning is a subset of machine learning that is often dubbed “learning to learn”. It generally refers to algorithms that learn from other algorithms, combining predictions and knowledge. It can also refer to a practitioner manually training a model with an algorithm that will at some stage be trained to the point of automation.

Natural Language processing

Natural Language processing or NLP is one of the most influential subsets of AI currently in use. It explores the concept of allowing a computer to understand text and speech in natural languages such as English and respond accordingly. There are 5 main components in NLP. Morphological and Lexical Analysis, Syntactic Analysis, Semantic Analysis, Discourse Integration, and Pragmatic Analysis.



Morphological and Lexical Analysis.

Morphological analysis refers to how words are formed and structured. This can be analysed to infer a words meaning, such as Unlikeable, because the word has the prefix un which means not, and the suffix able, to refer to the state of being. Therefore, like is a positive, if something is possesses a likeable quality, it is a positive for that thing, however, if it is unlikable then that isn’t positive.

Lexical Analysis instead of looking for a suffix or prefix of a word, looks for the words that came before or after it in a sentence in order to make sense of the overall context the word is used in. It can also gather words into groups or n-grams and compare them to other words and n-grams to garner the overall context.

Syntactic Analysis

Syntactic Analysis allows the program to assess the arrangement of words in a sentence so that they can be grammatically understood. It essentially formats based on whether or not the natural language input is aligned with major grammatical rules. For example, Microsoft words built in spell check has this feature. Most commonly used syntactic analysis techniques include Lemmatization, Parsing, and Word Segmentation. Lemmatization in essence, finds less complex ways of expressing words or phrases, and it uses a dictionary to step back as far as possible to find the root of a word. Parsing is what analyses the grammatical validity of a sentence, and can be used to explore the relationship between a word and the sentence they are contained in. Finally, Word segmentation is the separation text, from continuous into single words. Word segmentation is incredibly helpful in most eastern countries that don’t use the English alphabet, as they do not use spaces to mark the end of a word and the start of a new word.

Semantic Analysis

Semantic Analysis is the level of processing that focuses on the interpretation of a full sentence by examining Entities such as people and places, Language from one to another, NLG which generates sentences based on what a user inputs, such as an AI powered chatbot, and NLU that analyses passages of text into structures that the computer can change and break down efficiently.

Discourse Integration

This is primarily used to see how sentences relate to each other due to the words they contain. For example, if in one sentence, a man called tom is talked about, and in the following sentence which does not contain the word Tom, but it does contain he, therefore both sentences are about Tom. These are called Anaphora relationships and they are incredibly important for deciphering large volumes of text or for dialogue.

Pragmatic Analysis

This is how NLP effectively retrieves information about the text being used practically. This is used to get a meaning from a word that might not have an initial meaning, or the meaning doesn’t make sense in the grander scope of the sentence. For example, “Will you crack open the window?”. The word crack means to break, but if thinking logically or pragmatically, the word used in this context means open. This is the type of analysis that pragmatic analysis carries out.

All of these various components come together to form the basis of NLP, and allow for some incredibly futuristic chat bots, translation tools, accessibility with text to speech, and much more.

Deep learning

Deep learning is a difficult aspect of AI to identify, some argue it is a subset of machine learning, some say it’s the natural evolution of machine learning, and some say it is a separate branch of AI. It is however important enough regardless of classification that it must be discussed on its own.

Deep learning was inspired by the way that the human brain works, using what’s called neural networks. These neural networks allow an AI program to simulate a brains ability to adapt and learn with huge amounts of data, although there are constraints in place as each program is only good at what it is instructed to do. Deep learning allows for the processing of completely raw data, unstructured, unorganised, and uncompressed. Images, text and more can all be interpreted no matter the format. For example, let’s say that we had a set of photos of different pets, and we wanted to categorize by “cat”, “dog”, “hamster”, et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. (IBM, 2020).

It can self-adjust and self-mediate choices in order to create the most accurate and reliable outcomes. This is often dubbed unsupervised learning, as it needs no human interference to calibrate or set parameters for it to work within. Basically, any dataset it works with can be completely unlabelled. There are two types of deep learning, CNN and RNN.

Convolutional neural networks primarily are found in computer vision and image recognition programs can detect patters of unlabelled images and sort them into what makes them similar, even capable of generating its own interpretation of what a certain subject looks like.

Recurrent neural networks are used in speech recognition and natural language and more. They are often used in virtual assistants like Alexa and Siri and primarily deal with sequential data. They are known for their ability to understand that the output of a certain program must be informed by the original input and that they are not necessarily two separate entities.

Deep learning applications are often seen in the service sector, such as Healthcare with the digitization of hospital records, Law enforcement indicating fraud, speech and facial recognition, Finance with predictive algorithms to risk assess and predict trade patterns, and customer service such as chatbots.

Robotics

Robotics is a field of AI that is heavily tied into mechanical and electrical engineering, which work together with AI to create Intelligent robots. A lot of robotics borrows from other aspects of AI such as computer vision or machine learning to carry out their tasks effectively. Smart robots such as assembly line robots, robots designed to operate outside of healthy human conditions like underwater and in space, and robots that carry out menial tasks such as smart vacuums and so on are all common place in our world right now. Some might not consider robotics a branch of Ai on its own, however they at the very least share an important crossover.

Diagram, venn diagram

Description automatically generated

Most Intelligent robots can act autonomously or at least semi autonomously and interact with their environment using a multitude of sensors and actuators in order to gain information about the world surrounding them.

1. Recommender Systems

Recommender Systems are an incredibly important application for AI and are seen in abundance in everyday life. Recommender systems can be found in popular sites such as Netflix, Twitter, Facebook, but are also used in advertising and Shopping websites. They are a subsect of machine learning and are programmed to predict various things about a user and how they interact with a specific dataset, using various algorithms. There are several different types of recommender system, but the three that will be compared today will be Popularity based, Content based, and Collaborative based.



Popularity based recommenders are more than likely the simplest out of the three. They focus on recommending what’s popular among the user base. For example, Twitter has an entire page dedicated to “Trending”. This means that the tweets with the most views, retweets, and likes will be shown at the top of this page. Medium.com shows how IMDB’s weighted average formula works and can be seen below.

Weighted Rating (WR)=[vR/(v+m)]+[mC/(v+m)]

were,

v is the number of votes for the movie.

m is the minimum votes required to be listed in the chart.

R is the average rating of the movie; and

C is the mean vote across the whole report.

This is an example of how popularity-based recommender systems work, as this filters through the list of data within the dataset to look for any popular movies that fit this list and displays them. The advantages of this type of recommender is that it is simple, easy to implement, and shows a selection of what the average Populus would enjoy. However, it is incredibly limited in what it is capable of and is not very tailored to the individual users tastes.

Content based recommenders are what will be discussed properly in the implementation portion of this report. In essence, a Content based recommender allows for a user to see and be recommended things that they may already have an interest in. For example, on Netflix if you have watched the Gotham TV show, you might enjoy the Dark Knight movie, or even less specifically, if you watched James’ Bond, you may like Taken, the Bourne Ultimatum, and Mission Impossible since you enjoy action movies. These are some of the most widespread recommender systems available due to their wide range of flexibility, how they tailor the likes of the user. However, they can be more complex to implement then a popularity-based recommender and need an incredibly robust dataset with a lot of information in order to make informed decisions about the user.

Collaborative based recommender systems are about recommending content, not necessarily based on the content you’ve previously enjoyed, but more in line with the content that users like you have enjoyed. This is interesting as with a content-based recommender, if the program doesn’t know the user is interested in something they won’t recommend it, however with collaborative the user may still get recommended it as similar users may enjoy it. The drawbacks however can also include poor recommendation quality due to data sparsity. If you don’t have many users or the users you do have each reflect your data in wildly differing ways, it can be difficult to make any appropriate recommendations.

1. Code

The task for this CA was to produce a simple content based recommender system. In order to gain knowledge on how to program a recommender system, I followed two Tutorials : <https://www.datacamp.com/community/tutorials/recommender-systems-python>

<https://www.linkedin.com/learning/building-a-recommendation-system-with-python-machine-learning-ai/content-based-recommender-systems?autoAdvance=true&autoSkip=false&autoplay=true&resume=false&u=56744273>

Both of these tutorials did a fantastic job of showing the basic skills needed to create a recommender system in python. Juptyer notebook was the environment used with libraries like pandas and numpy installed to assist. The recommender system created for this CA was a system for recommending Appstore games, such as chess and sudoku. It is content based, and will calculate a similarity score for all entries in the dataset to search and find games that are similar to games that are input.

First we import pandas and load the dataset. Pandas is a library that allows python devs to use it for data analysis. For this recommender system the Appstore games dataset from Kaggle was chosen, as it has a wide range of different games and also each one has a unique id which a lot of Kaggle datasets were missing.

Text

Description automatically generated with medium confidence

Next, we use the TFIDF or Term Frequency-Inverse Document Frequency. This is used to carry a semantic meaning from the words that are carried in the dataset allowing them to relate to each other effectively. We first import it using the Scikit learn then we remove any stop words to make the process more efficient and straightforward. We also replace all of the null values with an empty string, and construct the matrix. Finally we output the relevant matrix’s shape.

Graphical user interface, text

Description automatically generated

Next we want to print out a number of descriptions in the matrix. We can do this by using whats known as the map function, which in essence, calls a specific part of each element, in this instance the ID, and printing out the associated Description to each id within the 50 spaces we have provided, between 3500 and 3550.

A picture containing text

Description automatically generated

Next comes the main portion of our recommender system. With the matrix created we can use the cosine similarity algorithm in order to calculate a value that shows the similarity score between two games.

A picture containing chart

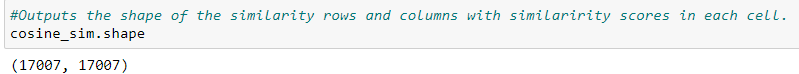
Description automatically generated

There are other algorithms such as the Pearson Correlation and Euclidean Distance. The Pearson correlation separates itself from the Cosine since its coefficient calculates the similarity between two jointly distributed random variables as opposed to two samples, whereas the Euclidean Distance looks at the distance between two points, it’s a little more streamlined but less scalable. For this example, we chose the Cosine Similarity although any of these would have worked, however the resulting similarity and suggestions may in-fact be different. We use this indirectly however as using sklearns linear kernel will output the same similarity score much quicker.

Graphical user interface, text

Description automatically generated

We then get the returned shape in order to quantify how many rows in each column we need to make the calculation, for each game to have a similarity score with another.



This is the first object in the cosine sim array, and below the individual similarity scores can be seen

Text

Description automatically generated with medium confidence

Next we will create a function that allows a user to input a game title and output a list of the 10 most similar game titles. This reverse maps the indexes in the metadata and the similarity scores and then outputs them.

Graphical user interface, text, application

Description automatically generated

We can then write our recommendation function. This needs to do a number of things, firstly retrieving the unique id of the game that matches the name, then looking through the similarity scores, make it a list of tuples where element 1 is the position and element two is the similarity score. Next we have to format the list based on similarity score, and get the top 10 elements of that list, ignoring the first as this is the initial element, and return the list to the viewer. Finally when we call this function, we pass in a game title, in the example below the game title is chess.

Text

Description automatically generated

Lets make another input with sudoku.

Text

Description automatically generated

As can be seen, as long as there is something in the dataset that has a similarity to something else that exists, then the recommender system will output any similar results.

1. Conclusion

Understanding AI is so important as a computing student, especially in this current climate where it is becoming more and more common place. Developing a recommender system is a great way to dip your toes into the topic for the first time from a development standpoint. It is simple and straightforward, however working with this technology opens up new doors that can lead down incredibly rewarding and complex paths.

1. References

West, D, Allen, J. (24th April 2018) How artificial intelligence is transforming the world.

Retrieved October 15th 2021 from <https://www.brookings.edu/research/how-artificial-intelligence-is-transforming-the-world/#:~:text=Summary,transforming%20every%20walk%20of%20life>

Russel, S. (June 2016) Should we fear supersmart robots?

Retrieved October 15th 2021 from <http://aima.cs.berkeley.edu/~russell/papers/sciam16-supersmart.pdf>

Joshi, N (2019) 7 Types of artificial Intelligence

Retrieved October 29th 2021 from <https://www.forbes.com/sites/cognitiveworld/2019/06/19/7-types-of-artificial-intelligence/?sh=2e2dd21b233e>

Nixon, M, Aguado, A. (2002) Feature Extraction and Image Processing.

Retrieved October 17th 2021 from

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.375.6848&rep=rep1&type=pdf>

IBM (2021) What is computer vision

Retrieved October 17th from

<https://www.ibm.com/topics/computer-vision>

Murphy, K (2012) Machine Learning: A Probabilistic Perspective

Retrieved October 24th from

<http://noiselab.ucsd.edu/ECE228/Murphy_Machine_Learning.pdf>

IBM(2020) What is Deep Learning?

Retrieved October 25th from

<https://www.ibm.com/cloud/learn/deep-learning>

Reddy, S(2020). Implementing a Recommendation System on IMDB Dataset through Machine Learning techniques.

Retrieved October 27th from

<https://medium.com/@sr7037/implementing-a-recommendation-system-on-imdb-dataset-through-machine-learning-techniques-47d0a86da9df>

Pierson, L(14th July 2017). Building a recommendation system with Python Machine Learning and AI. From

<https://www.linkedin.com/learning/building-a-recommendation-system-with-python-machine-learning-ai/content-based-recommender-systems?autoAdvance=true&autoSkip=false&autoplay=true&resume=false&u=56744273>

Sharma, A (29th May 2020). Beginner Tutorial: Recommender Systems in Python.

From

<https://www.datacamp.com/community/tutorials/recommender-systems-python>