A formal proposal for a MSc project that will be submitted in partial fulfillment of a University of Greenwich Master’s Degree

**“Product Matching”**

**Name: Wajahat Ali Mughal**

**Student ID: 001191074**

**Course of Study: MSc Computer Science**

**Date Proposal Submitted: 27/5/2022**

**Submission Date: 16/9/2022**

**Supervisor: Hai Huang**

**Topic Area:** Data Science & Machine Learning

**Keywords associated with the project:** Product matching, Word embedding, cosine similarity, Word2Vec, Support Vector Machine.

# MSc Modules studied that contribute towards this project: Applied Machine Learning, Enterprise Software Engineering Development

# Overview

The percentage of persons who have placed a purchase for a product or service online has dramatically increased during the past decade. Between the years 2010 and 2019, the percentage of total retail sales that were accounted for by e-commerce sales in the United States tripled, reaching a level of 12 percent(Bureau, 2022). It is impossible to ignore the significance of the impact that machine learning and artificial intelligence have in this context. Over seventy percent of the tech executives polled believe that artificial intelligence would have a beneficial impact on the retail sector, and eighty percent believe that AI will allow businesses to maximise their revenues(Edelman, 2019). Machine learning is typically applied to the personalization of services, the management of interactions with clients, and enterprise resource planning at this time(Ismail, 2015).

The difficulty of selecting a portfolio of products to provide to clients in order to achieve maximum income is equivalent to optimizing the selection of those products (Bernstein, 2015). In a similar vein, price optimization refers to the process of determining the optimal pricing strategy in order to maximise either revenue or profit on the basis of various models that predict customer demand(Fujimaki, 2016). Both are absolutely necessary in order to achieve a position of competitive advantage in the market. It is necessary to match products that are available in the offerings of competitors with those that are sold by the merchant being studied so that the aforementioned difficulties can be resolved. This kind of product matching is not a simple task because it requires utilizing the information that is offered by a variety of different sellers. The degree to which it is accurate and comprehensive varies. As a result, algorithms that are built need to be able to handle a variety of data types, as well as missing values and imprecise or incorrect entries.

1. **Methodology**

# The proposed system consists of a number of different modules, including Data preprocessing, word embedding, classification, and Evaluation. A feature selection and cleaning strategy will be conducted utilizing data science libraries as part of the data preprocessing step. The suggested model makes use of word2vec for word embedding; this helps to extract additional features from the input, which in turn helps to increase the model's accuracy. After that, the features are extracted, and a classification model is applied to determine whether or not the product being offered is a match for the extracted features. Because of this, a support vector machine, or SVM, is utilized. This type of machine works better when there is a bigger margin separating two classes; furthermore, because our dataset is tiny, training it will not require extra time. Calculating the precision and f1score are the final steps in the model evaluation process.

# 3. Objectives

To make consistent comparison framework to compare and evaluate record linkage models.

To find an evaluation of state-of-the-art record linkage models on product data.

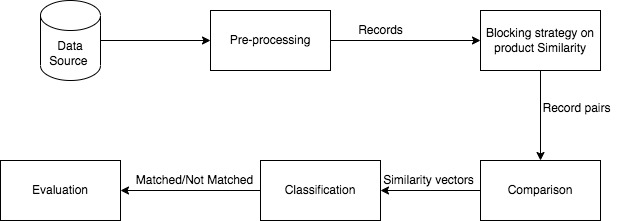
To build general approach for identifying whether traditional machine learning or deep learning techniques are suitable for a given set of data.

To Classify the Matched and Not Matched products from the Product offers.

To find Similarity vectors using the Blocking Strategy by using the record pairs.

# How the objectives will be achieved

To achieve these objectives, proposed model is integrated with the following modules and the algorithms, illustrated in the below figure.



**Fig: Proposed Flow of the Model**

Following dataset is used to train the model.

<https://drive.google.com/file/d/1S4SwciKmhw4xIZVQPr6fFSaxf6/edit>

The Dataset has 1500 product data which is in JSON format which contains different features for each product i.e., some products have key value pairs and some have not.

The Common Features of the dataset are Listed below:

Id, Category, title, description, brand, price and pair\_id.

**Data Preprocessing:**

This technique is helpful for the detection as well as the correction of the corrupt data, and it also identifies the erroneous data that may be present in the given dataset. This procedure not only modifies but also deletes and replaces any coarse data that may have been present in the dataset.

**Blocking Strategy on Product Similarity:**

Clustering is an important technique that is utilized in many different subfields of research, such as data mining, image retrieval, bio-computing, and many more. When clustering data points, distance metric is a very significant factor to consider. The most difficult part of the process is determining which distance metric to use for a particular dataset.

The selection of a distance measure, which will decide how similar the clusters will be, is an essential stage in any clustering(Vimal, 2008) of these two constituents is determined. This will have an effect on the shape of the clusters because some of the elements may be closer to one another according to one distance and farther away according to another.

It is reasonable to anticipate that the distance between things that are part of the same cluster will be quite short, whereas the distance between objects that are part of separate clusters will be rather long. In this article, we examine and contrast a variety of distance measurements.

In the research on data clustering, a great number of different distance measures have been presented. The Manhattan distance, the Minkowski distance, and the Hamming distance are all examples of metric functions that can be used to measure this. Other popular methods for determining distance include the Jaccard index, the cosine similarity, and the dice coefficient. Specific distance functions have been proposed for use with non-numerical datasets. For instance, edit distance is a well-known distance measure for text properties.

In this section, we will provide a concise explanation of the various distance metrics that are often employed.

**Euclidean Distance:**

The Euclidean distance, also known as the Euclidean metric, is the standard distance that may be determined by using a ruler to determine the distance between two points. It is the distance that can be travelled in a straight line between two places.

**Manhattan Distance:**

This can readily be extended to cover a wider range of dimensions. Integrated circuits frequently make use of the Manhattan distance, even though their wires only ever run in parallel to the X or Y axis. It is sometimes referred to as Minkowski's distance and rectilinear distance (Ester, 1996) City block, L1 distance, or taxi cab distance.

**Bit-Vector Distance:**

The Mb matrix, which has dimensions N by N, is computed. Every point has d dimensions, and the Mb (Pi, Pj) vector is determined using d-bits. This vector can be obtained in the following manner: If indeed the numerical value of the xth dimension of point is greater than the numerical value of the xth dimension of point pj, y, then the bit x of Mb (Pi, Pj) is set to 1 and the bit x of Mb (Pi, Pi) is set to 0. In other words, if the calculated value of the xth dimension of point is greater than the numerical value of the xth dimension of point pj, y, then the bit x After that, each of Mb's bit vectors is transformed into an integer value.

**Hamming Distance:**

The number of locations on two strings that are the same length but have different matching symbols is the Hamming distance between them. Let x, y A^n. The number of points in space where x and y diverge from one another is what we mean when we talk about the Hamming distance between two numbers, which is given by the notation dH (x, y). According to Xu (2005), the Hamming distance can be seen as the number of bits that must be altered or corrupted in order to transform one string into another. It is possible to utilize the number of characters rather than the number of bits in some situations. One way to interpret the Hamming distance between bit vectors is as the Manhattan distance.

**Cosine Similarity:**

It is a way to assess the degree of similarity between two vectors with n dimensions by determining the angle that exists between them. It is most commonly used in text mining to compare different publications (Gunawan, 2018). The cosine similarity is calculated based on two vectors of qualities, A and B.

When performing text matching, the tf-idf vectors of the documents are typically used to construct the attribute vectors A and B. Since the angle, is within the range of [0, π], the resulting similarity will yield the value of as meaning completely opposite, the value π / 2 as meaning independent, and the value 0 as meaning exactly the same, with values in between suggesting intermediate similarities or dissimilarities.

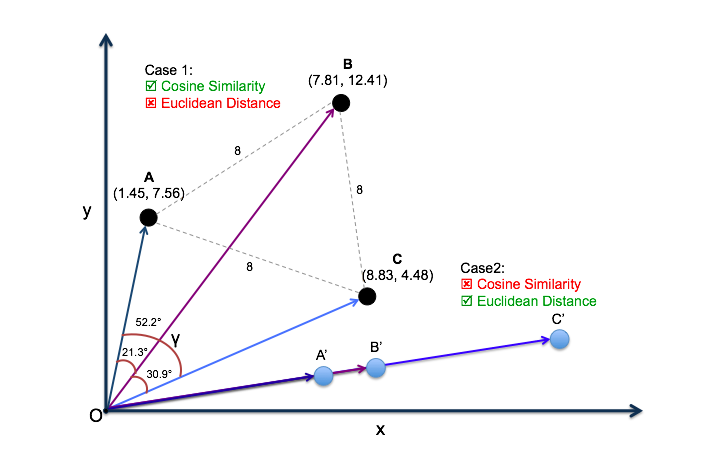
In the following section, we will discuss the concepts of Cosine similarity and Euclidean distance, as well as the several contexts in which these concepts might be utilized.

A measure of the degree of similarity between two non-zero vectors of an inner product space is referred to as the cosine similarity, and it is determined by the cosine of the angle that separates them. The cosine of an angle of 0 degrees is 1, and the cosine of any other angle in the range [0, π] radians is less than 1. It is therefore a judgement of orientation rather than magnitude: two vectors that have the same orientation have a cosine similarity of 1, two vectors that are oriented at 90 degrees relative to each other have a similarity of 0, and two vectors that are diametrically opposed have a similarity of -1, regardless of the magnitude of either vector. (Gunawan, 2018).

First, let's take a look at two scenarios in which one of the two measures (namely, cosine similarity or Euclidean distance) is more accurate than the other.

**Case 1: When Cosine Similarity is better than Euclidean distance**

Let's pretend that OA, OB, and OC are three vectors, just how they are shown in figure 1. The triangle with equilateral sides is formed by the points A, B, and C. This indicates that the Euclidean distance between these places (AB = BC = CA) is equal to one another. In this particular scenario, the Euclidean distance will not be an accurate metric to use for determining which of the three vectors are most comparable to one another. In spite of the fact that the

magnitudes (lengths) of the vectors are distinct from one another, the cosine similarity measure reveals that OA is closer to OB than it is to OC.

**Figure1: Illustration of the Cosine Similarity and Euclidean Distance**

**Case 2: When Euclidean distance is better than Cosine similarity**

Take into consideration an additional scenario in which the points A', B', and C' are coincident with one another, as shown in figure 1. In this particular scenario, the cosine similarity of all three vectors—OA', OB', and OC'—is identical to one another (equals to 1). The Euclidean measure of distance, on the other hand, will be more accurate, and it will reveal that A' is more closely related to B' than it is to C'.

**When to Use Cosine?**

When it is irrelevant to consider the magnitude of the vectors being compared, cosine similarity is typically applied as a metric for distance measurement. One situation in which this can occur is when working with text data that is represented by word counts. When a term, such as "science," appears in document 1 more frequently than it does in document 2, we have reason to believe that the former contains more information pertinent to the latter's subject matter about science. On the other hand, it's possible that the documents we're dealing with are of varying lengths, so keep that in mind (Wikipedia articles for example). Because document 1 was so much lengthier than document 2, it's likely that there were greater scientific developments in document 1. The cosine similarity adjustment makes up for this (Gunawan, 2018).

The most common scenario in which this statistic would be useful is when applied to text data. However, you may also wish to apply cosine similarity for other circumstances where certain aspects of the instances make it so that the weights might potentially be bigger without the results signifying anything different. One such illustration of this would be sensor data that were recorded throughout a range of intervals (in terms of time) between individual incidents.

***Even if two similar documents are geographically separated by a great distance according to the Euclidean distance because of their size (for example, the word "cricket" appeared 50 times in one document and 10 times in another), they may still have a smaller angle between them if the cosine similarity is used. This is an advantage of the cosine similarity. When the angle is reduced, the degree of resemblance increases.***

**Word Embedding:**

A word embedding is a form of learnt representation for text in which words that have the same meaning have a representation that is similar to one another. It is possible that this method of encoding words and documents will be recognized as one of the most significant advances in the application of deep learning to difficult challenges in natural language processing (Wang, 2020).

The continuous skip-gram model acquires knowledge by making predictions about the words that are adjacent to the current word. In a nutshell, the Continuous Skip-Gram Model forecasts words that come before and after the current word in the same phrase that fall within a predetermined range (Xiong, 2019). The morphology of a word is not taken into account by the skip-gram model. Even though they share some lemmas, two words are considered to be their own unique entities. This results in the learning of vectors for the words that are related to one another due to the fact that they appear in comparable contexts. As an illustration, the term "bank" can refer to both a financial organization and to land along a river.

On the other side, Word2vec is an approach to natural language processing that was introduced in the year 2013. The word2vec algorithm is based on a neural network model, which it employs in order to learn word connections from a vast body of text (Ma, 2015).

The two-layer neural network known as Word2vec is responsible for processing text by "vectorizing" the words. It takes a text corpus as its input and outputs a set of vectors as its result. Words in the corpus represented by feature vectors. Feature vectors. After being trained, such a model is able to recognize terms that are synonymous with one another and suggest further words for a sentence that is incomplete. As its name suggests, the word2vec programmed associates each individual word with a specific collection of numbers that is referred to as a vector. The vectors have been carefully selected in such a way that a straightforward mathematical function, known as the cosine similarity between the vectors, may be used to determine the degree of semantic similarity between the words that are represented by those vectors (Ma, 2015).

In comparison to the bag of words and the IF-IDF system, Word2Vec has a number of distinct advantages. Word2Vec is able to preserve the semantic meaning of a document's myriad of individual words. The information within the context is not lost. The fact that the size of the embedding vector may be kept relatively minimal is another significant benefit of the Word2Vec technique. (Ma, 2015).

**Classification:**

In comparison to the bag of words and the IF-IDF system, Word2Vec has a number of distinct advantages. Word2Vec is able to preserve the semantic meaning of document’s myriad of individual words. The information within the context is not lost. The fact that the size of the embedding vector may be kept relatively minimal is another significant benefit of the Word2Vec technique.

The technique of supervised learning known as the decision tree can be applied to problems involving both classification and regression, but in practice, it is most often utilized for the resolution of classification issues**.**

In this approach, a group of training instances is segmented into ever-smaller subgroups while concurrently an associated decision tree is constructed in an incremental fashion. When the learning process is complete, a decision tree that encompasses the training set is returned (Patel, 2018).

Utilizing a decision tree to segment the data space into clustered (also known as dense) regions and empty (also known as sparse) regions is the primary concept here.

In Decision Tree Classification, a new example is categorized by first putting it through a battery of tests, which together establish the example's label for the class to which it belongs. The results of these examinations are presented in the form of a hierarchical structure known as a decision tree. The Divide-and-Conquer Algorithm is what Decision Trees are based on (Patel, 2018).

The fact that decision trees are inherently unstable makes them one of the least desirable options for use in classification methods. This means that even a seemingly insignificant shift in the data can have a significant impact on the form that the best decision tree takes. They are frequently wrong to a relatively small degree. There are a great number of different predictors that perform better with comparable data (Patel, 2018).

On the other hand, Support Vector Machine, also known as SVM, is an algorithm for supervised machine learning that may be utilized for classification as well as regression. Despite the fact that we also discuss regression problems, its use is more appropriate for classification. The goal of the Support Vector Machine (SVM) algorithm is to locate a hyperplane in an N-dimensional space that can classify the data points in a separate manner.

The support vector machine (SVM) transforms your input using a method known as the kernel trick, and then, based on these transformations, it identifies an ideal boundary between the various outputs that can be generated (Evgeniou, 2001).

The use of SVM has a number of potential benefits, one of which is that it functions rather well in situations in which there is a distinct gap separating different classes. In high-dimensional spaces, SVM performs better than other methods. When there are more samples than dimensions, supervised learning methods like support vector machines (SVM) are useful. SVM is memory efficient to a reasonable degree (Evgeniou, 2001).

1. **Problem Areas covered by the Proposed System**

Instead of using Euclidean distance, which corresponds to the L2 norm of difference between vectors and therefore means that distances computed might be skewed depending on the units of the features, the proposed model uses cosine similarity because it can find similarity for the synonyms that are used in product categories by using the angle that is formed between two vectors that are projected into a multi-dimensional space. This is done in place of using Euclidean distance. (Gunawan, 2018).

Word2Vec is used in the proposed model because it maintains the semantic meaning of different words in a document, the context information is not lost, and the size of the embedding vector is very small. This solves the problem caused by the skip-gram model, which tries to predict several context words from a single input word, which takes more time to train. Word2Vec is used in the proposed model because it retains the semantic meaning of different words in a document. (Ma, 2015).

In the proposed system, support vector machines (SVM) are used to solve the problems that are caused by the decision tree algorithm. These problems include the fact that the algorithm causes instability even when only a small change is made to the data, that it requires more time when there are more features, and that calculations can become more complicated in such a scenario. In order to overcome these challenges, an algorithm that is productive in high-dimensional spaces and that can perform very well even if the decision border that divides two classes is greater needs to be developed. (Evgeniou, 2001).

# Legal, Social and Ethical Issues

**Can it be that some people might find the product offensive?**

This product is intended for use in product matching, and it does so by employing machine learning techniques and utilizing historical data in a way that does not violate any laws and does not utilize any content that is offensive to users.

**Will you be using copyright material? How will you get permission to use it?**

In order to prevent any disagreements in the future, the research work done by the many authors was taken into consideration and mentioned in the appropriate manner.

**Does the product adhere to accepted standards and is it secure enough to hold data? In case you are dealing with confidential data have you seek advice from the Ethics Committee?**

The user will be prompted to grant permission for the system to use their information, and a permission mechanism will already be in place to store user information in a database.

**Is there a chance that there might be health and safety issues with the product or with the project in general?**

Considering that this product is nothing more than a digital platform that helps match different products in the most effective way possible, its use poses no risk whatsoever.

**Is there a chance that there might be accessibility issues with the product?**

For the user to have the best possible experience with the product, this system needs that they maintain a constant internet connection. In the event of a cyberattack or a natural disaster, there is always a proxy server available, and the vendor encrypts all of the data using various cryptographic algorithms, such as SHA (Secure Hash Algorithm).

**Will you need permits to record video/audio/photography?**

The only permission that is required is authorization to store and access the user's personal data, and this is the only permission that is required. Access to this system does not require any other permissions to be granted.

**Will you need model release forms?**

Because the code is held in private possession by the vendor, there is no requirement that the model release form be used for this system. Additionally, there is no requirement that the code be made available in the public domain. In the event that the suggested procedures are published, I will be required to submit an application for the material's copyright.

**How does the data protection act affect you?**

Failure to comply with regulations can result in a notice of the enforcement being issued, which not only halts the processing of data but also prohibits enterprises from functioning normally. In some cases, hefty fines are also levied on firms for failing to comply. Individually, each member of the organization's employees, including managers and directors, has an equal share of responsibility for any instances of non-compliance.

**How does the Disability discrimination act affect you?**

There is nothing within this system, not even the code itself, that violates the provisions of the disability discrimination act in any way.

1. **Key work**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | task | Activities | estimated time | status |
| Data  Pre-Processing | Data manipulation using NumPy and pandas | To clean and pre-process the data before giving it to the model. | 3 Weeks | Research is completed |
|  | Normalization | To process of translating data into the range [0, 1] (or any other range) or simply transforming data onto the unit sphere. | Research is completed |
|  | Feature Selection | reducing the input variable to your model by using only relevant data and getting rid of noise in data | Pending |
| Blocking | Naive approach | To record linkage and compare every record in the one dataset to every record in the other dataset. | 4 weeks | Research is completed |
| Comparison | Count Vectorizer | Single axis has to compare only one attribute value from each record, sometimes it may be helpful to compare multiple attributes to compute a single numeric similarity value. | 2 weeks | Research is completed |
| Classification | Matched/Not Matched | Each record pair is predicted to match or not match using its feature vector. | 2 weeks | Pending |
| Evaluation | Precision | Measure the performance of the Classifier. | 2 weeks | Pending |

# 7. Resources

Bernstein, 2015. Dynamic assortment customization with limited inventories.. *Manufacturing and Service Operations Management,* pp. 538-553.

Bureau, U. C., 2022. *QUARTERLY RETAIL E-COMMERCE SALES,* washington: US Census Bureau.

Edelman, 2019. *Edelman AI Survey.,* s.l.: Whitepaper.

Ester, 1996. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,* 8 Miinchen, Germany: Institute for Computer Science, University of Munic.

Evgeniou, 2001. *Support Vector Machines: Theory and Applications.* s.l., Machine Learning and Its Applications, Advanced Lectures.

Fujimaki, 2016. Large-scale price optimization via network flow. *Advances in Neural Information Processing Systems,* Volume 29, pp. 3855-3863.

Gunawan, 2018. The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *Journal of Physics Conference Series ,* Volume 1, p. 978.

Ismail, 2015. Data mining in electronic commerce: Benefits and challenges. *International Journal of Communications, Network and System Sciences,* pp. 1-16.

Ma, 2015. *Using Word2Vec to process big text data.* San Jose, CA, Using Word2Vec to process big text data.

Patel, 2018. Study and Analysis of Decision Tree Based Classification Algorithms. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING,* 6(10), pp. 74-78.

Vimal, 2008. *An Experiment with Distance Measures for Clustering,* Hyderabad - 500 032, INDIA: International Institute of Information Technology.

Wang, 2020. *A Comparative Study on Word Embeddings in Deep Learning for Text Classification.* s.l., 4th International Conference on Natural Language Processing and Information Retrieval.

Xiong, 2019. New Generation Model of Word Vector Representation Based on CBOW or Skip-Gram. *Computers, Materials and Continua ,* 58(2), pp. 259-273.

Xu, 2005. “Survey of Clustering Algorithms”. *IEEE Transactions on Neural Networks,* 16(3), pp. 1-8.