Chapter 6
Recommendation Machine Learning Model for Data Centre Design

6.1 Introduction

Machine Learning Model is a subset of Artificial Intelligence and is expected to optimize every facet of future data center operations, including planning and design, managing IT workloads, ensuring uptime, and controlling costs. Machine learning includes supervised, unsupervised, and reinforced learning techniques. A machine learning model is the output of the training process and is defined as the mathematical representation of the real-world process. This existing data is used by Machine learning algorithms to develop predictive models and automate several time-consuming tasks.

Supervised Learning

In Supervised Learning, a machine is trained using ‘labelled’ data. Datasets are said to be labelled when they contain both input and output parameters. In other words, the data has already been tagged with the correct answer.

Supervised machine learning is immensely helpful in solving real-world computational problems. The algorithm predicts outcomes for unexpected data by learning from labelled training data. Therefore, it takes highly-skilled data scientists to build and deploy models.

Different Types of Supervised Learning: Regression, Classification, Naive Bayesian Model, Random Forest Model, Neural Networks, Support Vector Machines, Forecasting, Assembling, etc.

Algorithms for Supervised Learning: Decision Trees, Naïve Bayes Classifier Algorithm (Supervised Learning - Classification), Logistic Regression (Supervised learning – Classification), Linear Regression (Supervised Learning/Regression), Support Vector Machine Algorithm (Supervised Learning - Classification), Decision Trees (Supervised Learning Classification/Regression), Random Forests (Supervised Learning Classification/Regression), Nearest Neighbours (Supervised Learning)

Unsupervised Learning

Unsupervised Learning uses machine learning algorithms to analyse and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Unsupervised learning models are utilized for three main tasks clustering, association, and dimensionality reduction. Unsupervised Learning Algorithms allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods.

Algorithms for Unsupervised Learning: K Means Clustering Algorithm (Unsupervised Learning - Clustering)

6.2 Predictive Modeling

Predictive modeling is a method of predicting future outcomes by using data modeling, big data incredible volumes of raw structured, semi-structured, and unstructured data is an unused resource of intelligence that can support business decisions and enhance operations. As data continues to diversify and change organizations are implementing predictive analytics.

A common misconception is that predictive analytics and machine learning are the same things. This is not the case. Predictive analytics and machine learning overlap. Predictive analytics contains a variety of statistical techniques, including machine learning, predictive modeling, and data mining, and uses statistics both historical and current to estimate, or 'predict', future outcomes.

There are two types of predictive models. They are Classification models, which predict class membership, and Regression models which predict a number.

Decision trees: Decision trees are a simple, but powerful form of multiple variable analysis. They are produced by algorithms that identify various ways of splitting data into branch-like segments. Decision trees partition data into subsets based on categories of input variables, helping you to understand someone’s path of decisions.

Regression (linear and logistic): Regression is one of the most popular methods in statistics. Regression analysis estimates relationships among variables, finding key patterns in large and diverse data sets and how they relate to each other.

6.3 Recommendation System

A recommendation system is a subclass of machine learning, a branch of artificial intelligence that generally deals with ranking or rating products/users. A recommender system is a system that predicts ratings/suggestions for a user-specific item. Present is an allowance for lots of portals to accumulate data from their users and use that data to predict the keenness on and aversions of their users [2].

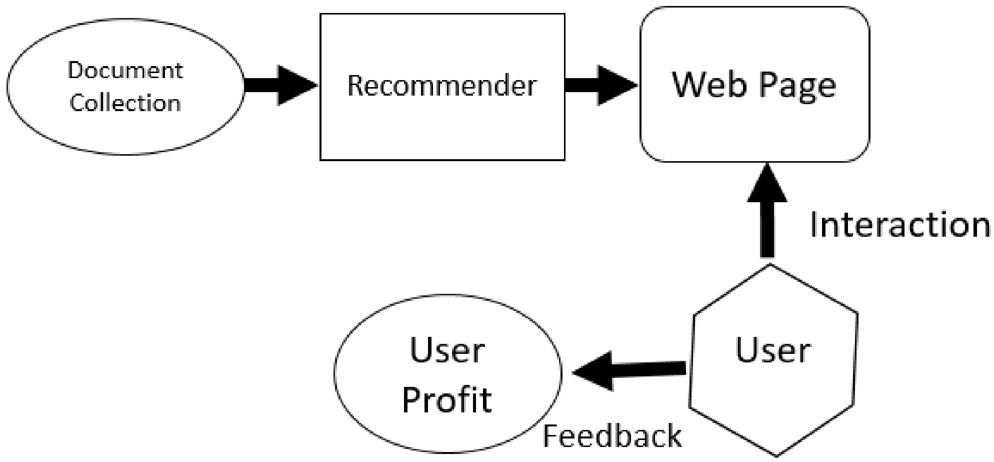

Figure 6.1: Recommendation Systems [2]

There are many different ways to build recommender systems algorithmic and formulaic, many Algorithms are approaches that can vary in complexity, but complexity does not translate to "good" performance. Simple formulaic implementations of recommendation engines to promote content also Reddit, Hacker News, and Google are using same. Recommender systems are used widely for recommending movies, articles, restaurants, places to visit, items to buy, etc.

Popularity-Based Recommendation System

This one category of recommendation system which mechanism on the standard of popularity and or everything which is in manner. These systems checked the product or films which are in inclination or are most popular among the users and straight recommend those.

Context-Aware Recommender Systems

A context-aware recommender system applies sensing and analysis of user context to provide personalized services. The contextual information can be driven by sensors to improve the accuracy of the recommendations.

Demographic-Based Recommendation System

The demographic Recommender system generates recommendations based on the user demographic attributes. It categorizes the users based on their attributes and recommends movies by utilizing their demographic data.

Utility-Based Recommendation System

Utility-based recommender systems provide recommendations based on the computation of the utility of each item for the user. Some utility-elicitation methods have been developed based on multi-attribute utility theory to represent a decision maker's complete preference.

Knowledge-Based Recommendation System

Knowledge-based recommender systems (knowledge-based recommenders) are a specific type of recommender system that is based on explicit knowledge about the item assortment, user preferences, and recommendation criteria (i.e., which item should be recommended in which context).

Collaborative Filtering

Collaborative Filtering recommends items based on comparison methods between users and/or objects. The simple assumption behindhand the algorithm is that users with comparable interests have mutual preferences.

Collaborative filtering is the process of predicting the interests of a user by identifying preferences and information from many users. There are two common types of approaches in collaborative filtering, memory-based, and model-based approaches.

Memory-Based Approaches: also often referred to as neighbourhood collaborative filtering. Essentially, ratings of user-item combinations are predicted based on their neighbourhoods. **User-based** essentially means that like-minded users are going to yield strong and similar recommendations. **Item-based** collaborative filtering recommends items based on the similarity between items calculated using user ratings of those items.

Model-Based Approaches: are predictive models using machine learning. Features associated with the dataset are parameterized as inputs of the model to try to solve an optimization-related problem using decision trees, rule-based approaches, latent factor models, etc. Collaborative filtering is a technique that can filter out items that a user might like based on reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user.

Content-Based Recommender Systems

A content-based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes action on the recommendations, the engine becomes more and more accurate.

Supervised machine learning algorithms are used to make a classifier to differentiate between interesting and unexciting items for the manipulator.

Content-based systems generate recommendations based on the user's preferences and profile. They try to match users to items that they've liked previously. Content-based models focus on the ratings provided by the target user themselves. Item-level data source needs a strong source of data associated with the attributes of the item. Item profile content-based recommendation algorithm method, the necessity to construct a profile for an individual item, which encloses the imperative properties of individual item using TF-IDF vectorizer algorithm. User-level data source, need some sort of user feedback based on the item you're providing recommendations. User profile vector that labels the user predilection during the making of the user's profile, we use utility conditions that describe the association amongst users and items [2].
The concepts of Term Frequency (TF) and Inverse Document Frequency (IDF) are used in information retrieval systems and also content-based filtering mechanisms (content-based recommender). They are used to determine the relative importance of a document/article/news item/movie etc.
Which contains the significant properties of each item and term-frequency can be calculated.

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}}$$

where f_{ij} is the frequency of term(feature) i in document(item) j.

- The inverse-document frequency can be calculated with:

$$IDF_i = log_e \frac{N}{n_i}$$

where, n_i number of documents that mention term i. N is the total number of docs.

- Therefore, the total formula is:

$$TF - IDFscore(w_{ij}) = TF_{ij} * IDF_i$$

Here, doc profile is the set of words with

Figure 6.2: TF-IDF Vectorizer [2]

6.4 Hybrid Recommendation System

Hybrid recommender systems are designed to use different available data sources to generate strong inferences. Hybrid recommendation systems have two predominant designs, parallel and sequential. The parallel design provides the input to multiple recommendation systems, and each of those recommendations is combined to generate one output. The sequential design provides the input parameters to a single recommendation engine, the output is passed on to the following recommender in a sequence.

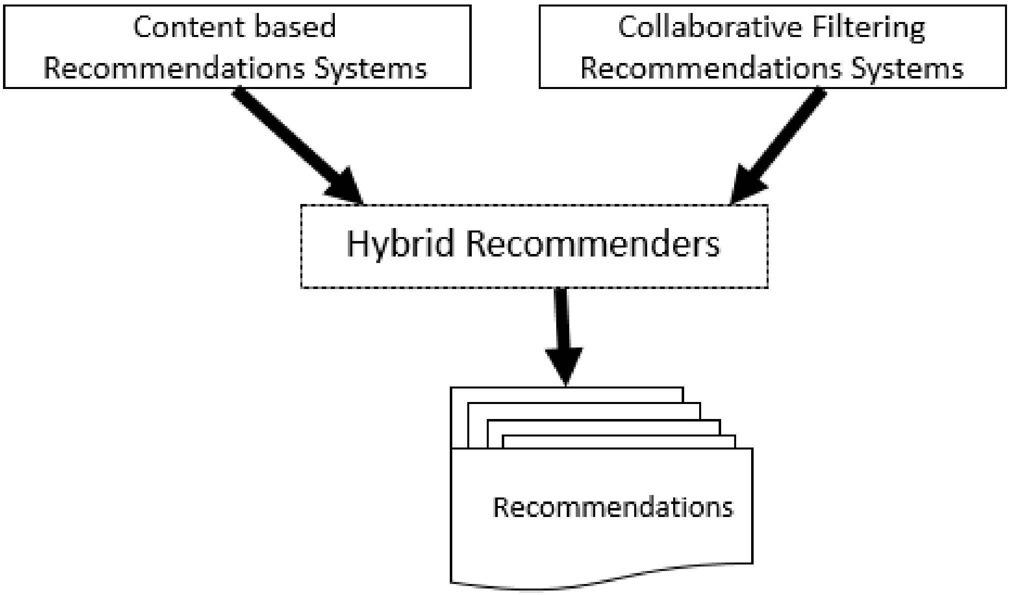

Figure 6.3: Hybrid Recommendation Systems

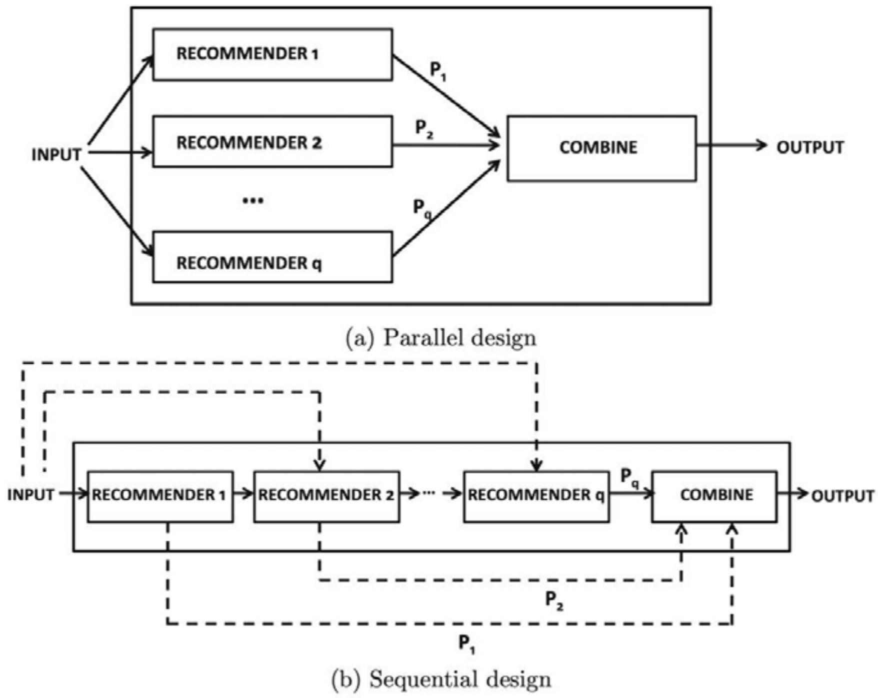

Figure 6.4: Parallel and Sequential Design

A hybrid recommendation system is a special type of recommendation system which can be considered as a combination of the content and collaborative filtering method.

Recommendation systems are generally used in a range of applications for recommending products or items to the user. There are two popular methods used for filtering the recommendations, content-based and collaborative filtering [4]. These methods face the issue when there is not enough data to learn the relation between users and items. In such cases, the third type of approach is used to build the recommendation system named as Hybrid Recommendation System.

Hybrid recommender system approaches can be implemented in various ways like by using content and collaborative-based methods to generate predictions separately and then combining the prediction or we can just add the capabilities of collaborative-based methods to a content-based approach [7].

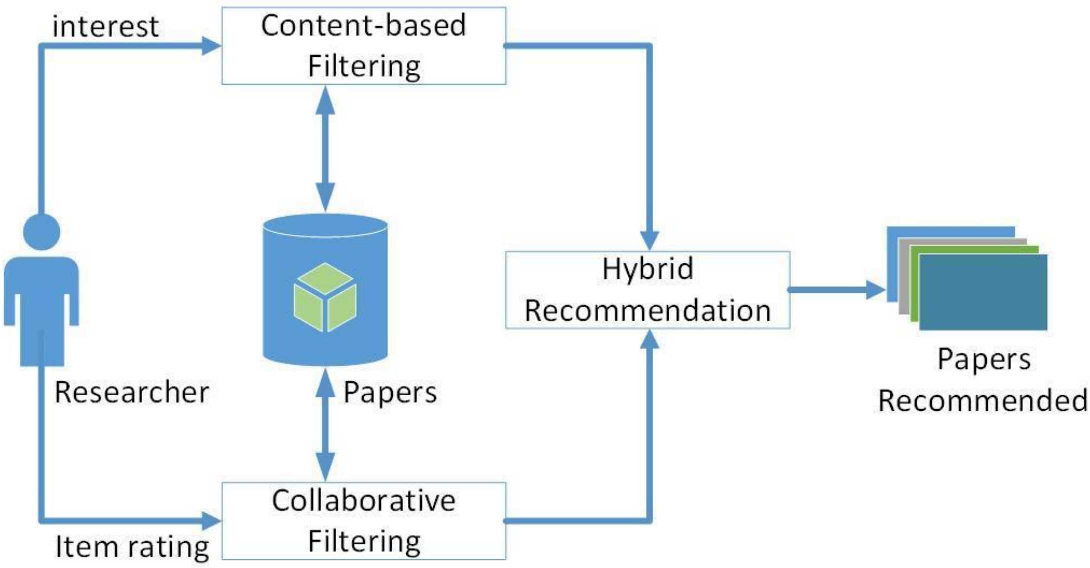

Figure 6.5: Collaborative-Based Methods to a Content-Based Approach

Following the approaches, we can divide the data into two types using which we can generate a recommendation system.

Explicit Feedback: the data which contains the user's explicit feedback. Explicit feedback can be a kind of rating from the user to the item which tells about the status of the user and whether he liked the product or not.

Implicit Feedback: this data is not about the rating or score which is provided by the user, it can be some information that can inform about clicks, watching movies, playing songs, etc. Accuracy is the fraction of correct recommendations out of the total possible. Recommender systems share several conceptual similarities with the classification and regression modeling problem.

6.5 K-means Clustering

K-means: K-means clustering is a very famous and powerful unsupervised machine learning algorithm. It is used to solve many complex unsupervised machine learning problems. Visualizing product clusters in a subset of data Fitting K-Means to the dataset K-means is an iterative algorithm that groups similar data into clusters. It calculates the centroids of k clusters and assigns a data point to that cluster having the least distance between its centroid and the data point [10].

K means it is an iterative clustering algorithm that helps you to find the highest value for every iteration. Initially, the desired number of clusters is selected. In this clustering method, you need to cluster the data points into k groups. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.

The output of the algorithm is a group of "labels." It assigns a data point to one of the k groups. In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

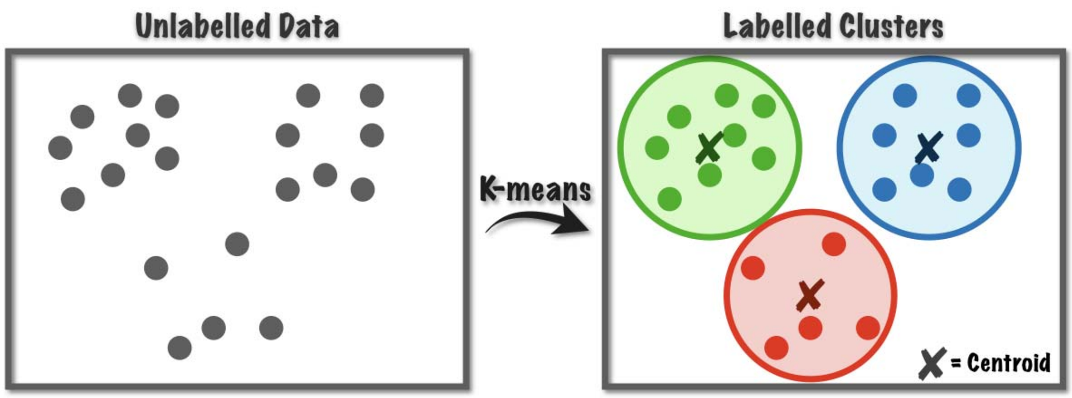

Figure 6.6: Steps of the K-means Algorithm Source [10]

We start by choosing a value of k. Here, let us say k = 3. Then, we randomly assign each data point to any of the 3 clusters. Compute cluster centroid for each of the clusters. The red, blue, and green stars denote the centroids for each of the 3 clusters. Next, reassign each point to the closest cluster centroid. In the figure above, the upper 5 points got assigned to the cluster with the blue centroid. Follow the same procedure to assign points to the clusters containing the red and green centroids.

Then, calculate centroids for the new clusters. The old centroids are grey stars; the new centroids are red, green, and blue stars. Finally, repeat steps 2-3 until there is no switching of points from one cluster to another. Once there is no switching for 2 consecutive steps, exit the K-means algorithm.

6.6 Nearest Neighborhood

The standard method of Collaborative Filtering is known as the Nearest Neighbourhood algorithm. There are user-based CF and item-based CF. Let's first look at User-based CF. We have an n × m matrix of ratings, with user u_i, i = 1,...n and item p_j, j=1, …m. Now we want to predict the rating r_{ij} if the target user I did not watch/rate an item j. The process is to calculate the similarities between target users I and all other users, select the top X similar users, and take the weighted average of ratings from these X users with similarities as weights [10].

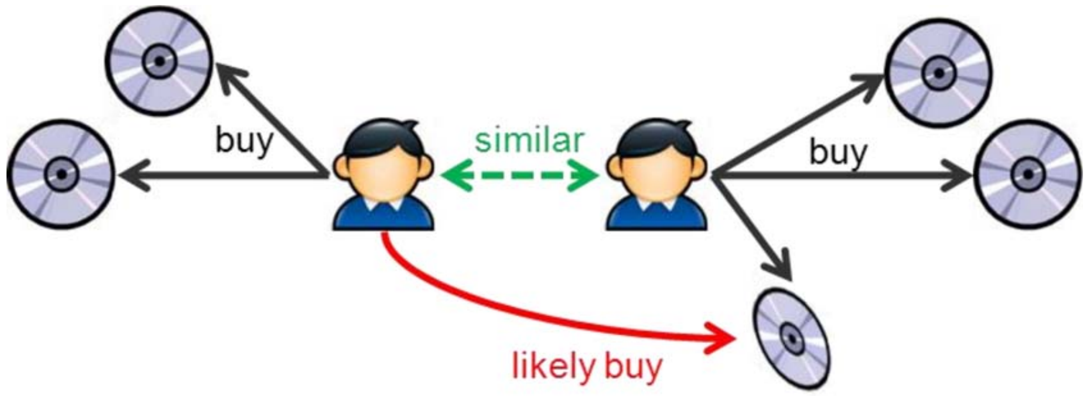

Figure 6.7: Nearest Neighbourhood [10]

While different people may have different baselines when giving ratings, some people tend to give high scores generally, and some are pretty strict even though they are satisfied with the items. To avoid this bias, we can subtract each user's average rating of all items when computing the weighted average, and add it back for the target user, shown as below.

$$r_{ij} = \bar{r}_i + \frac{\sum_k Similaries(u_i, u_k)(r_{kj} - \bar{r}_k)}{number\ of\ ratings} \qquad r_{ij} = \frac{\sum_k Similaries(u_i, u_k) r_{kj}}{number\ of\ ratings}$$

Figure 6.8: Nearest Neighbourhood Similarity

6.7 Cosine Similarity

Cosine of the angle between the two vectors of the item, vectors of A and B is calculated for imputing similarity. If the vectors are closer, then small will be the angle and large will be the cosine.

$$\text{Similarity}(X,Y) = \frac{X.Y}{|X| \times |Y|}$$

Figure 6.9: Cosine Similarity [9]

There is individuality for considerable of the operator's data. Item pieces of information permit us to start open-handed recommendations to users. A content-based recommender machine does not be depending on the user's data, accordingly unfluctuating if a new-fangled user arises, we can recommend the handler as stretched as we have the user data to build his profile. It makes sure of not to agonize from an unfriendly start. Items statistics ought to be in respectable volume. Structures should be offered to compute the similarity. Techniques used to measure Similarity: Pearson correlation (CORR), Cosine (COS), adjusted cosine (ACOS), constrained correlation (CCORR), Mean Squared Differences (MSD), Euclidean (EUC) [9].

6.8 Matrix Factorization

The matrix factorization is Singular Value Decomposition (SVD). Since sparsity and scalability are the two biggest challenges for the standard CF method, it comes to a more advanced method that decomposes the original sparse matrix into low-dimensional matrices with latent factors/features and less sparsity. That is Matrix Factorization [10].

The first thing to understand is Singular Value Decomposition (SVD). Based on Linear Algebra, any real matrix R can be decomposed into 3 matrices U, Σ, and V. Continuing using the movie example, U is an $n \times r$ user-latent feature matrix, and V is an $m \times r$ movie-latent feature matrix. Σ is an $r \times r$ diagonal matrix containing the singular values of the original matrix, simply representing how important a specific feature is to predict user preference [9].

$$R = U\Sigma V^T$$

$$U \in IR^{n\times r}, \quad \Sigma \in IR^{r\times r}, \quad V \in IR^{r\times m}$$

Figure 6.10: Matrix Σ to First k Dimensions [9]

To sort the values of Σ by decreasing the absolute value and truncating matrix Σ to the first k dimensions (k singular values), we can reconstruct the matrix as matrix A. The selection of k should make sure that A can capture the most of variance within the original matrix R, so that A is the approximation of R, A ≈ R. The difference between A and R is the error that is expected to be minimized. This is exactly the thought of Principle Component Analysis.

When matrix R is dense, U and V could be easily factorized analytically. However, a matrix of movie ratings is super sparse. Although there are some imputation methods to fill in missing values, we will turn to a programming approach to just live with those missing values and find factor matrices U and V. Instead of factorizing R via SVD, we are trying to find U and V directly with the goal that when U and V multiplied back together the output matrix R' is the closest approximation of R and no more a sparse matrix. This numerical approximation is usually achieved with Non-Negative Matrix Factorization for recommender systems since there are no negative values in ratings.

6.9 Vector Space Model

In this model, each item is stored as a vector of its attributes (which are also vectors) in an n-dimensional space and the angles between the vectors are calculated to determine the similarity between the vectors. Next, the user profile vectors are also created based on his actions on previous attributes of items, and the similarity between an item and a user is also determined similarly [9].

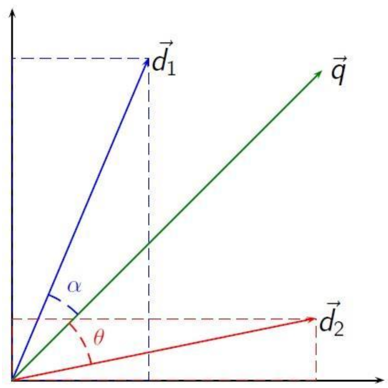

Figure 6.11: Vector Space Model [9]

Machine learning is hundreds of algorithms out of hundreds the recommendation systems are strongly recommended or used when we work on product/item or user-based requirements. I have used the recommendation system here. If I apply every algorithm it will take a lot of time. Choosing the right algorithm is linked up with the problem statement. It can save both money and time. So, it is important to know what type of problem we are dealing with.

The town data center facility is proposed so user rating and review are not in our database so this is a fresh item so we need to item to item base model. I have proposed the town data center and his facility is defined in a paragraph each paragraph has more than 100 words and it's very difficult for each word and facility present in table format. So we define only two columns the first is product_uid and the second is product_description. The CVS comma-separated values file has two columns and five rows. It has 8242 words.

Item to item based recommendation system based on product description Applicable when a business is set up for the first time, from the plot, we can see that there is a lot of overlap between the data points. A decision tree or Random Forest works on the principle of non-linear classification. We can use it if some of the data points are overlapping with each other.

Many algorithms work on the assumption that classes can be separated by a straight line. In such cases, Logistic regression or a Support Vector Machine should be preferred. It easily separates the data points by drawing a line that divides the target class. Linear regression algorithms assume that data trends follow a straight line. These algorithms perform well in the present case.

Split the data into train and test. Now we can proceed by applying Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine algorithms to check the training time for a classification problem. I have concluded my analysis by selecting the correct machine learning algorithm. Furthermore, it is always advisable to use two algorithms for addressing the problem statement. This could provide a good reference point for the audience.

6.10 Model 1 Item-Based Hybrid Recommendation System

Item to item base hybrid recommendation system based on the product description Predicting clusters based on key search words. I used this as applicable when a business is set up for the first time, and no rating by the customer. This model used K-means algorithms to unsupervised machine learning to cluster data with labels and then vectorization recommendation fetch the supervised labelled data. All the practical work was done on google colab lab.

 # First I have imported all required libraries

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

from sklearn.neighbors import NearestNeighbors

from sklearn.cluster import KMeans

from sklearn.metrics import adjusted_rand_score

 # After it i will proceed by reading the csv file

product_descriptions = pd.read_csv('/content/product_descriptions.csv')

product_descriptions.shape

 # Checking for missing values

product_descriptions = product_descriptions.dropna()

product_descriptions.shape

product_descriptions.head()

index	product_uid	product_description
0	100001	Edge data centre latency has always been a problem for data center management critical concern due to big data, the Internet of Things, cloud and streaming services, and other technology trends. End users and devices demand anywhere, anytime access to applications, services, and data housed in today's data centers, and latency is no longer tolerable. The high-performance and cost-effective way to provide customers with content and functionality. Local Edge data centers are placed near the areas they serve and are managed remotely. Small Edge data centers have the same components as a traditional data center but are packed into a much smaller footprint. ISO 27001, ISO 9000, SSAE, HIPPA, Availability, reliability, Tier 3-4 as per rack cluster vised, Space 2500 square fit, more than 1mega watt power capacity, primary optical fibre connectivity. professional remote services to end user, monitoring level 3
1	100002	Town data centers are defined differently by different data center professionals based on their roles, industries, or priorities, and due to the relative infancy of town data centers as an established trend. Local town data centers are placed near the industrial areas they serve and are managed remotely. Town data centers have the same components as a traditional data center or server room but are packed into a much smaller footprint. Mission critical town data center manage application or services based. secure location with geographic stability, natural disaster & man-made secure, High-definition video surveillance of both the interior and exterior with archival support, Live technical monitoring by expert NOC staff, 24/7/365 support from a live expert, redundancy, ISO 27001, ISO 9000, Availability, reliability, Tier 1-2-3-4 as per rack cluster vised, location Metoda GIDC, Space 1500 square fit, 600-kilowatt power capacity, primary optical fibre connectivity, 1Gb fast internet. Professional services to end user, monitoring level 2. N+1 base level of resources required for the system functionality plus a single backup/ redundancy.
2	100003	Server room critical computational resources are maintained in an environment that protects them both during normal operation, as well as in the event of power failures, fires,

		floods, and other emergency conditions. One rack and ten by ten room size or nearest no-redundancy and can't perform data center standard. N refers to the minimum number of resources (amount) required to operate an IT system. No redundancy solution is available for the system. In case of a failure until the issue is diagnosed and resolved. no one certificate and maintenance cost is high
3	100004	On-premises Data Center is almost server room type but treat as a data centre critical computational resources are maintained in an environment that protects them both during normal operation, as well as in the event of power failures, fires, floods, and other emergency conditions. One rack and ten by ten room size or nearest no-redundancy and can't perform data center standard. N refers to the minimum number of resources (amount) required to operate an IT system. No redundancy solution is available for the system. In case of a failure until the issue is diagnosed and resolved. maintenance cost is high
4	100005	Cloud Data Centre is work as virtualization the location is remote side. Cloud data centre twice the required resources/capacity. Plus single backup is an additional redundancy step. Or Three of the required resources/capacity. Plus twice backup as an additional redundancy step. Active. In an active configuration, the redundant component is operated simultaneously with the original component. However, in case of the original fails, the redundant component will be used. Passive. In a passive configuration, the redundant component is available yet not operational while the original component is active. It will be activated to provide functionality in the event of a failure. Load sharing (standby). This full fills the availability gap until the original or active component is completely available. Additionally, load sharing can be used here as a partial or a temporary redundancy method to provide additional capacity. ISO 27001, ISO 20000-1, SSAE 18 SOC 1 Type II, SOC 2 Type II and SOC 3, HIPAA, TIA-942, SAS 70, PCI-DSS 3.2, prices are more than any one data centre
5	100006	Co-Location Data center work as a server room or local data center or server its connection VPN based the service provider provides the location and maintenance and monitoring and services also as per agreement the hardware provide us or rental basis the terms and facility are also as per standard cloud data center. We rental some space for our requirement. Additionally, load sharing can be used here as a partial or a temporary redundancy method to provide additional capacity. ISO 27001, ISO 20000-1, SSAE 18 SOC 1 Type II, SOC 2 Type II and SOC 3, HIPAA, PCI-DSS 3.2, prices are more than any one data center. Cost is higher than virtual server rentals.

Table 6.1: Item-Based Hybrid Product Description

product_descriptions1 = product_descriptions.head(500)

 # Product_descriptions1.iloc[:,1]

product_descriptions1["product_description"].head(10)

Output:

0 Edge data centre the latency has always been a...
1 Town data centers are defined differently by differ...
2 Server room critical computational resources a...
3 On-premises Data Center it's almost server room...
4 Cloud Data Centre is work as virtualization...
5 Co-Location Data center work as a server room...

Name: product_description, dtype: object

#TF-IDF VECTORIZER

 # Feature extraction from product descriptions converting the text in product description into numerical data for analysis.

vectorizer = TfidfVectorizer(stop_words='english')
X1 = vectorizer.fit_transform(product_descriptions1["product_description"])
X1

Output:

<6x223 sparse matrix of type '<class 'numpy. float64'>
With 380 stored elements in Comp ressed Spars Row layout>

K-means clustering is a very famous and powerful unsupervised machine learning algorithm. It is used to solve many complex unsupervised machine learning problems

 # Visualizing product clusters in a subset of data

 # Fitting K-Means to the dataset

```
X=X1
kmeans = KMeans(n_clusters = 6, init = 'k-means++')
y_kmeans = kmeans.fit_predict(X)
plt.plot(y_kmeans, ".")
plt.show()
def print_cluster(i):
  print("Cluster %d:" % i),
  for ind in order_centroids[i, :6]:
     print(' %s' % terms[ind]),
  print
```

 # Optimal clusters is

```
true_k = 6
model = KMeans(n_clusters=true_k, init='k-means++', max_iter=100, n_init=1)
model.fit(X1)
KMeans(max_iter=100, n_clusters=6, n_init=1)
```

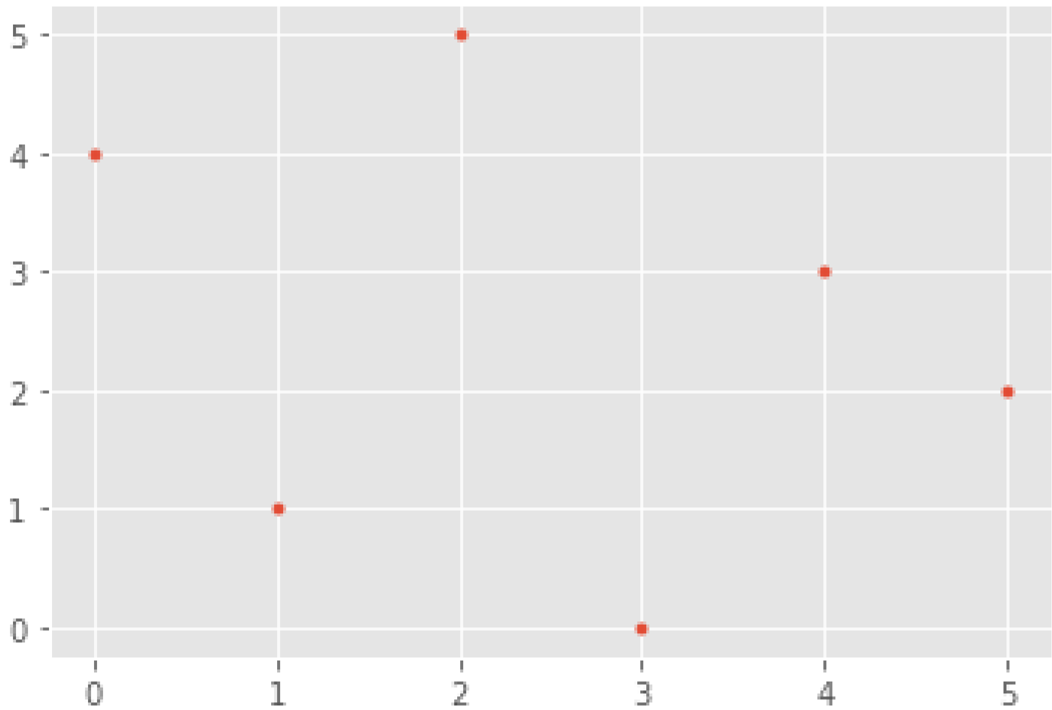

Figure 6.12: Product-Based Cluster

Recommendation of product based on the current product selected by a user.

To recommend a related product based on, frequently bought together.

Top words in each cluster based on the product description

#Vector Space Model:

```
print("Top terms per cluster:")
order_centroids = model.cluster_centers_.argsort()[:, ::-1]
terms = vectorizer.get_feature_names_out(input_features=None)
for i in range(true_k):
    print_cluster(i)
```

Output

Top terms per cluster:

Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:
soc	resources	town	component	Data	data
provide	room	centers	active	Edge	room
server	certificate	data	original	centers	resources

data	protects	support	redundant	services	center
ii	operation	secure	soc	latency	treat
center	number	expert	additional	End	premises

Table 6.2: Cluster-Based Predicting

```
# Predicting clusters based on key search words
def show_recommendations(product):
   #print("Cluster ID:")
   Y = vectorizer.transform([product])
   prediction = model.predict(Y)
   #print(prediction)
   print_cluster(prediction[0])
print(' \n Predicting clusters based on key search words \n')
# Keyword: town
show_recommendations("town")
```

Output

Predicting clusters based on key search words

Cluster 4:
town
centers
data
support
secure
expert

Table 6.3: Predicting Clusters Based on key Search Words

In case a word appears in multiple clusters, the algorithm chooses the cluster with the highest frequency of occurrence of the word. Once a cluster is identified based on the user's search words, the recommendation system can display items from the corresponding product clusters based on the product descriptions. This works best if a business is setting up its e-commerce website for the first time and does not have user-item purchase/rating history to start with initially. This recommendation system will help the users get a good recommendation to start with and once the buyers have a purchased history, the recommendation engine can use the model-based collaborative filtering technique.

6.11 Model 2 Cosine Similarity Unsupervised Recommendation Systems

Generate a term frequency-inverse document frequency (TF-IDF) matrix of unigrams, bigrams, and trigrams for individual sources. The stop words the TF-IDF module to disregard mutual English words. Then we calculate the comparison between all products using SciKit Leanr's linear_kernel this is equal to cosine similarity. Repeat over and done with individually item's similar items and store the hundreds of most similar.

 # First I have imported all required libraries

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import linear_kernel

 # After it will proceed by reading the csv file

ds = pd.read_csv("/content/sample-data.csv")

index	id	Description
0	1	Edge data centre latency has always been a problem for data center management critical concern due to big data, the Internet of Things, cloud and streaming services, and other technology trends. End users and devices demand anywhere, anytime access to applications, services, and data housed in today's data centers, and latency is no longer tolerable. The high-performance and cost-effective way to provide customers with content and functionality. Local Edge data centers are placed near the areas they serve and are managed remotely. Small Edge data centers have the same components as a traditional data center but are packed into a much smaller footprint. ISO 27001, ISO 9000, SSAE, HIPPA, Availability, reliability, Tier 3-4 as per rack cluster vised, Space 2500 square fit, more than 1mega watt power capacity, primary optical fibre connectivity. professional remote services to end user, monitoring level 3
1	2	Town data centers are defined differently by different data center professionals based on their roles, industries, or priorities, and due to the relative infancy of town data centers as an established trend. Local town data centers are placed near the industrial areas they serve and are managed remotely. Town data centers have the same components as a traditional data center or server room but are packed into a much smaller footprint. Mission critical town data center manage application or services based. secure location with geographic stability, natural disaster & man-made secure, High-definition video surveillance of both the interior and exterior with archival support, Live technical monitoring by expert NOC staff, 24/7/365 support from a live expert, redundancy, ISO

		27001, ISO 9000, Availability, reliability, Tier 1-2-3-4 as per rack cluster vised, location Metoda GIDC, Space 1500 square fit, 600-kilowatt power capacity, primary optical fibre connectivity, 1Gb fast internet. Professional services to end user, monitoring level 2. N+1 base level of resources required for the system functionality plus a single backup/ redundancy.
2	3	Server room critical computational resources are maintained in an environment that protects them both during normal operation, as well as in the event of power failures, fires, floods, and other emergency conditions. One rack and ten by ten room size or nearest no-redundancy and can't perform data center standard. N refers to the minimum number of resources (amount) required to operate an IT system. No redundancy solution is available for the system. In case of a failure until the issue is diagnosed and resolved. no one certificate and maintenance cost is high
3	4	On-premises Data Center is almost server room type but treat as a data centre critical computational resources are maintained in an environment that protects them both during normal operation, as well as in the event of power failures, fires, floods, and other emergency conditions. One rack and ten by ten room size or nearest no-redundancy and can't perform data center standard. N refers to the minimum number of resources (amount) required to operate an IT system. No redundancy solution is available for the system. In case of a failure until the issue is diagnosed and resolved. maintenance cost is high
4	5	Cloud Data Centre is work as virtualization the location is remote side. Cloud data centre twice the required resources/capacity. Plus single backup is an additional redundancy step. Or Three of the required resources/capacity. Plus twice backup as an additional redundancy step. Active. In an active configuration, the redundant component is operated simultaneously with the original component. However, in case of the original fails, the redundant component will be used. Passive. In a passive configuration, the redundant component is available yet not operational while the original component is active. It will be activated to provide functionality in the event of a failure. Load sharing (standby). This fulfils the availability gap until the original or active component is completely available. Additionally, load sharing can be used here as a partial or a temporary redundancy method to provide additional capacity. ISO 27001, ISO 20000-1, SSAE 18 SOC 1 Type II, SOC 2 Type II and SOC 3, HIPAA, TIA-942, SAS 70, PCI-DSS 3.2, prices are more than any one data centre
5	6	Co-Location Data center work as a server room or local data center or server its connection VPN based the service provider provides the location and maintenance and monitoring and services also as per agreement the hardware provide us or

		rental basis the terms and facility are also as per standard cloud data center. We rental some space for our requirement. Additionally, load sharing can be used here as a partial or a temporary redundancy method to provide additional capacity. ISO 27001, ISO 20000-1, SSAE 18 SOC 1 Type II, SOC 2 Type II and SOC 3, HIPAA, PCI-DSS 3.2, prices are more than any one data center. Cost is higher than virtual server rentals.

Table 6.4: Sample-Data File

 # Step 1 Train the engine

```
tf = TfidfVectorizer (analyzer = 'word', ngram_range = (1, 3), min_df = 0, stop_words='english')
tfidf_matrix = tf.fit_transform(ds['description'])
```

Cosine Similarity:

```
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
results = {}
for idx, row in ds.iterrows():
    similar_indices = cosine_similarities[idx].argsort()[:-100:-1]
    similar_items = [(cosine_similarities[idx][i], ds['id'][i]) for i in similar_indices]
    # initially item is the itself, so remove it.
    # Each wordlist entry is like: [(1,2), (3,4)], with each tuple being (score, item_id)
    results[row['id']] = similar_items[1:]
    print('done!')
```

Output:

done!

 # Step 2 Predict

```
# This function to get a responsive item name from the description field, given an item ID
def item(id):
    return ds.loc[ds['id'] == id]['description'].tolist()[0].split(' - ')[0]
# just reads the results out of the dictionary. No real logic here.
def recommend(item_id, num):
    print("Recommending " + str(num) + " products similar to " + item(item_id) + "...")
    print("-------")
    recs = results[item_id][:num]
    for rec in recs:
```

print("Recommended: " + item(rec[1]) + " (score:" + str(rec[0]) + ")")

 # Just plug in any item id here (1 to 500), and the number of recommendations wants (1 to 99)

 # I can get a list of valid item IDs by evaluating the variable 'ds', or a few are listed below

recommend(item_id=4, num=5)

done!

Recommending 5 products similar to On-premises Data Center...

Recommended: Server room…………………………(score:0.8374724070580033)

Recommended: Co-Location Data center…………...(score:0.08894589762065501)

Recommended: Town data centers…………………..(score:0.06992894129913639)

Recommended: Edge data…………………….……..(score:0.05461144937840369)

Recommended: Cloud Data Centre………………...(score:0.051568266588612334)

6.12 Model 3 Content-Based Supervised Recommendation System

This model is an algorithm used for the Content-Based Recommendation System it also supervised machine learning algorithms, Classification output.

 # First I have imported all required libraries

import numpy as np

import pandas as pd

import sklearn

import matplotlib.pyplot as plt

import seaborn as sns

from tabulate import tabulate

import warnings

warnings.simplefilter(action='ignore', category=FutureWarning)

 # After it will proceed by reading the csv file.

ratings = pd.read_csv("/content/d4.csv")

ratings.head()

Index	type	erp	tier	criticality	sla	s_level	dr	price
0	Small	Tally	1	Low	99.67	Up to 3	NO	1000
1	Small	Tally	2	Mid	99.74	Up to 4	NO	2000
2	Small	Tally	3	High	99.98	Up to 6	YES	3000
3	Small	Tally	4	High	99.99	Up to 9	YES	4000
4	Medium	CMS	1	Low	99.67	Up to 3	NO	2000

5	Medium	CMS	2	Mid	99.74	Up to 4	NO	3000
6	Medium	CMS	3	High	99.98	Up to 6	YES	4000
7	Medium	CMS	4	High	99.99	Up to 9	YES	5000
8	Large	SAP	1	Low	99.67	Up to 3	NO	5000
9	Large	SAP	2	Mid	99.74	Up to 4	NO	7000
10	Large	SAP	3	High	99.98	Up to 6	YES	9000
11	Large	SAP	4	High	99.99	Up to 9	YES	11000
12	Small	Miracle	1	Low	99.67	Up to 3	NO	1000
13	Small	Miracle	2	Mid	99.74	Up to 4	NO	2000
14	Small	Miracle	3	High	99.98	Up to 6	YES	3000
15	Small	Miracle	4	High	99.99	Up to 9	YES	4000
16	Small	Shree	1	Low	99.67	Up to 3	NO	1000
17	Small	Shree	2	Mid	99.74	Up to 4	NO	2000
18	Small	Shree	3	High	99.98	Up to 6	YES	3000
19	Small	Shree	4	High	99.99	Up to 9	YES	4000
20	Medium	UMS	1	Low	99.67	Up to 3	NO	2000
21	Medium	UMS	2	Mid	99.74	Up to 4	NO	3000
22	Medium	UMS	3	High	99.98	Up to 6	YES	4000
23	Medium	UMS	4	High	99.99	Up to 9	YES	5000
24	Large	HANA	1	Low	99.67	Up to 3	NO	5000
25	Large	HANA	2	Mid	99.74	Up to 4	NO	7000
26	Large	HANA	3	High	99.98	Up to 6	YES	9000
27	Large	HANA	4	High	99.99	Up to 9	YES	11000

Table 6.5: Item Server Details

```
# Store Labelled data in variable
n_ratings = len(ratings)
n_type = len(ratings['type'].unique())
n_erp = len(ratings['erp'].unique())
n_tier = len(ratings['tier'].unique())
n_criticality = len(ratings['criticality'].unique())
n_sla = len(ratings['sla'].unique())
n_s_level = len(ratings['s_level'].unique())
n_dr = len(ratings['dr'].unique())
n_price = len(ratings['price'].unique())
print(f"\n ... Content-Based Recommendation System... \n ")
print(f"Total Option : {n_ratings}")
print(f"ERP Services Type : {n_type}")
print(f"ERP Options : {n_erp}")
print(f"Number of Tier: {n_tier}")
```

print(f"Criticality Numbers : {n_criticality}")

print(f"SLA ratings: {n_sla}")

print(f"Security Levels : {n_s_level}")

print(f"Remote backup Option : {n_dr}")

print(f"Price Options : {n_price}")

…Output…

... Content-Based Recommendation System...

index	Services	Options
0	Total Option	28
1	ERP Services Type	3
2	ERP Options	7
3	Number of Tier	4
4	Criticality Number	3
5	SLA ratings	4
6	Security Levels	4
7	Remote backup Option	2
8	Price Options	8

Table 6.6: Content-Based Recommendation System Output

user_freq = df[['type', 'price']].groupby('type').count().reset_index()

user_freq.columns = ['Services', 'Options']

user_freq.head()

print(f" \n Number of Services Type : {user_freq}")

index	Services	Options
0	Large	8
1	Medium	8
2	Small	12

Table 6.7: Services wise Output

print(" \n Select operation. \n ")

print("1...Small")

print("2...Medium")

print("3...Large")

print("\n")

while True:

 # Take input from the user

choice = input("Enter Your Choice (1 / 2 / 3): ")

 # Check if the choice is one of the three options

if choice in ('1', '2', '3'):

if choice == '1':

```
        print(df.loc[df['type'] == 'Small'])
      elif choice == '2':
        print(df.loc[df['type'] == 'Medium'])
      elif choice == '3':
        print(df.loc[df['type'] == 'Large'])
      # Check if a user wants another Option
      # Break the while loop if the answer is no
      next_option = input(" \n Let's do next Option ? (yes / no): ")
      if next_option == "no":
       break
    else:
      print("Invalid Input")
```

Selection Option.

1………Small

2……..Medium

3……….Large

Enter Your Choice (1 / 2 / 3) : 2 "If the User Enter Two then output is…."

index	type	erp	tier	criticality	sla	S_level	dr	price
4	Medium	CMS	1	Low	99.67	Up to 3	NO	2000
5	Medium	CMS	2	Mid	99.74	Up to 4	NO	3000
6	Medium	CMS	3	High	99.98	Up to 6	YES	4000
7	Medium	CMS	4	High	99.99	Up to 9	YES	5000
20	Medium	UMS	1	Low	99.67	Up to 3	NO	2000
21	Medium	UMS	2	Mid	99.74	Up to 4	NO	3000
22	Medium	UMS	3	High	99.98	Up to 6	YES	4000
23	Medium	UMS	4	High	99.99	Up to 9	YES	5000

Table 6.8: User input wise Output

If users are satisfied with this output then type no and break the loop,

Let's do the next Option ? (Yes / no): no

If users are required to another option then type yes and enter a choice like 2 then output is as per input.

6.13 Model 4 Random Forest is a Supervised Machine Learning Algorithm

Random forest is a Supervised Machine Learning Algorithm that is Machine Learning Regression Problem in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. First import our data and analyses the data.

 # First I have imported all required libraries

import numpy as np

import pandas as pd

 # After it will proceed by reading the csv file.

data=pd.read_csv("/content/s3.csv")

data.head()

Output:

Index	Table	Details	QTY	Unit	Price
0	Office Furniture	Computer Table	4	nos	31892
1	Office Furniture	Computer Table	3	nos	20097
2	Office Furniture	Storage Cabot	2	nos	36866
3	Office Furniture	Storage Cabot	1	nos	11071
4	Office Furniture	Admin Cabot	1	nos	14954
5	Office Furniture	Meeting Room	1	nos	16950
6	Office Furniture	Revolving Chair	7	nos	39452
7	Office Furniture	Visiting Chair	4	nos	6836
8	IT Infra Office Equipment's Setup	OSP(valid 20 years)	1	nos	15000
9	Server details and cost	Power Vault ME4024 Storage Array	1	nos	900000
10	IT Infra Office Equipment's Setup	All In One PC (8GB RAM, 512SSD, i3 Processor, 4core)	3	nos	143400
11	IT Infra Office Equipment's Setup	LED Monitor (8GB RAM, 512SSD, i3 Processor, 4core)	5	nos	45000
12	IT Infra Office Equipment's Setup	Desktop (16GB RAM, 512SSD, i3 Processor, 4core)	4	nos	180000
13	IT Infra Office Equipment's Setup	Laptop (8GB RAM, 512SSD, i3 Processor, 4core)	3	nos	187200
14	IT Infra Office Equipment's Setup	L2 Manage Switch	1	nos	18500
15	IT Infra Office Equipment's Setup	Wi-Fi Access Point	2	nos	13200
16	IT Infra Office Equipment's Setup	Biometric Door Lock	1	nos	9900

17	IT Infra Office Equipment’s Setup	Biometric Attendance Machine up to 20 User	1	nos	13000
18	IT Infra Office Equipment’s Setup	NVR 16 Cha with 30 Days Backup (4TB -2 HDD)	1	nos	28500
19	IT Infra Office Equipment’s Setup	EPBX	1	nos	35750
20	IT Infra Office Equipment’s Setup	Phone	10	nos	12800
21	IT Infra Office Equipment’s Setup	55" TV	2	nos	101400
22	IT Infra Office Equipment’s Setup	Video Conferencing Systems	1	nos	72600
23	IT Infra Office Equipment’s Setup	D Link Router for Office Internet	1	nos	1800
24	IT Infra Office Equipment’s Setup	Multifunction Printer	1	nos	21300
25	IT Infra Office Equipment’s Setup	LED Light	24	nos	50880
26	IT Infra Office Equipment’s Setup	5 KVA Online UPS	1	nos	100000
27	Data Center Air Cooling cost	Split AC 2 ton	1	nos	45500
28	Data Center Air Cooling cost	Exos fane	2	nos	2560
29	Data Center Air Cooling cost	Pedestal FAN	2	nos	5198
30	Data Center Air Cooling cost	Voltas 2 ton Duct AC	8	nos	551920
31	Data Center Air Cooling cost	Voltas 2 ton Tower AC	2	nos	125780
32	Data Center Air Cooling cost	Blue star 4 ton Tower AC	2	nos	229980
33	IT Infra Office Cabling and Floor charges	CAT6a LAN Cable Laying with Casing Capping	500	mtr	30000
34	IT Infra Office Cabling and Floor charges	IO with Plate (LAN and Intercom)	20	nos	3000
35	IT Infra Office Cabling and Floor charges	9U Rack	1	nos	4700
36	IT Infra Office Cabling and Floor charges	IP Dome 2MP Camera inbuilt Audio	8	nos	28000
37	IT Infra Office Cabling and Floor charges	IP Bullet 2MP Camera inbuilt Audio	2	nos	7500

38	IT Infra Office Cabling and Floor charges	Antivirus	10	nos	16000
39	IT Infra Office Cabling and Floor charges	Tile Floor 2X2	1200	sqft	294000
40	IT Infra Office Cabling and Floor charges	POP 2X2	1200	sqft	80400
41	IT Infra Office Cabling and Floor charges	L3 PoE 24 port Switch for P to P Device	1	nos	160980
42	IT Infra Office Cabling and Floor charges	42U Network and Firewall Rack	1	nos	33950
43	Space for Data Centre	Purchase Space 1500 square feet	1	nos	9750000
44	Data Centre Power Supply Setup Cost	PGVCL Supply 150kva	1	nos	1080000
45	Data Centre Power Supply Setup Cost	Solar Setup 50kva	1	nos	2050000
46	Data Centre Power Supply Setup Cost	Diesel Generator 150kva	1	nos	3600000
47	Data Center 40 Rack Setup Cost	42U Server RACK	40	nos	1358000
48	Data Center 40 Rack Setup Cost	L3 24 port Switch	40	nos	4135440
49	Data Center 40 Rack Setup Cost	15 KVA Online UPS	40	nos	11784000
50	Data Center 40 Rack Setup Cost	Power Redundancy Switch	40	nos	2632000
51	Data center server cost	dell/hp	800	nos	214909760
52	Yearly Budget	Maintenance & Up gradation	1	year	9000000
53	Yearly Budget	Contingency	1	year	100000
54	Yearly Budget	Miscellaneous IT	1	year	100000
55	Yearly Budget	Miscellaneous Admin	1	year	50000
56	IT Infra Office Equipment's Setup	ISO,27001:2013,9001:2015,	2	year	20000
57	IT Infra Office Equipment's Setup	LAN State Network Monitoring Cost, 1000 user	1	year	45878
58	IT Infra Office Equipment's Setup	Firewall SOPHOS XG 750	1	year	600000
59	Technical Employee	Manager	1	year	480000
60	Technical Employee	Receptionist cum Admin	1	year	192000

61	Technical Employee	Server/Network/Firewall Expert	3	year	1080000
62	Technical Employee	Technical Support	2	year	480000
63	Technical Employee	Technical Field Expert	4	year	960000
64	Internet Leased Line and PRIL Facility	40Mbps ILL for DC (1:1)	1	year	144000
65	Internet Leased Line and PRIL Facility	100Mbps Broad Band Connection with Static IP	1	year	30000
66	Internet Leased Line and PRIL Facility	PRIL	1	year	84000
67	Data Centre Power Supply Setup Cost	PGVCL Supply 600kva	1	year	953700
68	Data Centre Power Supply Setup Cost	Diesel Generator 600kva	1	year	142402
69	Software	Software Cost	100	year	6150000
70	Software	VMWare Cost	100	year	4411900

Table 6.9: Data Center Equipment and Related Materials Cost

data.describe()

Output:

index	QTY	Price
count	71.0	71.0
mean	59.647887323943664	3944660.5070422534
std	225.05060374385317	25495993.72699288
min	1.0	1800.0
25%	1.0	20048.5
50%	2.0	80400.0
75%	4.5	515960.0
max	1200.0	214909760.0

Table 6.10: Converting Tax Data to Digit

data.info()

Output:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 71 entries, 0 to 70

Data columns (total 5 columns):

 # Column Non-Null Count Dtype

--- ------ -------------- -----

0 Table 71 non-null object

1 Details 71 non-null object

2 QTY 71 non-null int64
3 Unit 71 non-null object
4 Price 71 non-null int64
dtypes: int64(2), object(3)
memory usage: 2.9+ KB

Encoding: **This is one of the most common things that we perform in datasets. In data science and machine learning, we use encoding to convert categorical data into integers. Here we using LabelEncoder.**

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data['Table']=le.fit_transform(data['Table'])
data['Details']=le.fit_transform(data['Details'])
data['QTY']=le.fit_transform(data['QTY'])
data['Unit']=le.fit_transform(data['Unit'])
data['Price']=le.fit_transform(data['Price'])
data.head()
```

Output:

index	Table	Details	QTY	Unit	Price
0	7	15	3	1	23
1	7	15	2	1	18
2	7	58	1	1	26
3	7	58	0	1	8
4	7	8	0	1	1

Table 6.11: Encoding data value Output

 # Building a Machine Learning Model, Here we perform tasks like splitting the data, applying Supervised Machine Learning Random Forest algorithms, and predicting the output. Placing Portion Variable to X and Target variable to y.

 # Train-Test-Split is performed

```
from sklearn.model_selection import train_test_split
x=data.drop(['Table','Price'],axis=1)
y=data['Price']
xr,xt,yr,yt=train_test_split(x,y,test_size=0.25)
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import
r2_score,mean_squared_error,mean_squared_log_error,make_scorer
```

```
mod=RandomForestRegressor(n_estimators=100)
model.fit(x,y)
yp=model.predict(xt)
print(r2_score(yt,yp))
print(mean_squared_error(yt,yp))
print(mean_squared_log_error(yt,yp))
```

Output:

```
0.8862585201793722
39.45565555555555
0.04885544751086378
```

6.14 Model 5 Content-Based Supervised Machine Learning Algorithm

Content Based Supervised Machine Learning Algorithm that is Machine Learning Classification problems. It builds decision trees on different samples and takes their majority vote for classification.

 # First I have imported all required libraries

```
import numpy as np
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

 # After it will proceed by reading the csv file

```
ratings = pd.read_csv("/content/d4.csv")
ratings.head()
```

Index	type	erp	tier	criticality	sla	s_level	dr	price
0	Small	Tally	1	Low	99.67	Up to 3	NO	1000
1	Small	Tally	2	Mid	99.74	Up to 4	NO	2000
2	Small	Tally	3	High	99.98	Up to 6	YES	3000
3	Small	Tally	4	High	99.99	Up to 9	YES	4000
4	Medium	CMS	1	Low	99.67	Up to 3	NO	2000
5	Medium	CMS	2	Mid	99.74	Up to 4	NO	3000
6	Medium	CMS	3	High	99.98	Up to 6	YES	4000

7	Medium	CMS	4	High	99.99	Up to 9	YES	5000
8	Large	SAP	1	Low	99.67	Up to 3	NO	5000
9	Large	SAP	2	Mid	99.74	Up to 4	NO	7000
10	Large	SAP	3	High	99.98	Up to 6	YES	9000
11	Large	SAP	4	High	99.99	Up to 9	YES	11000
12	Small	Miracle	1	Low	99.67	Up to 3	NO	1000
13	Small	Miracle	2	Mid	99.74	Up to 4	NO	2000
14	Small	Miracle	3	High	99.98	Up to 6	YES	3000
15	Small	Miracle	4	High	99.99	Up to 9	YES	4000
16	Small	Shree	1	Low	99.67	Up to 3	NO	1000
17	Small	Shree	2	Mid	99.74	Up to 4	NO	2000
18	Small	Shree	3	High	99.98	Up to 6	YES	3000
19	Small	Shree	4	High	99.99	Up to 9	YES	4000
20	Medium	UMS	1	Low	99.67	Up to 3	NO	2000
21	Medium	UMS	2	Mid	99.74	Up to 4	NO	3000
22	Medium	UMS	3	High	99.98	Up to 6	YES	4000
23	Medium	UMS	4	High	99.99	Up to 9	YES	5000
24	Large	HANA	1	Low	99.67	Up to 3	NO	5000
25	Large	HANA	2	Mid	99.74	Up to 4	NO	7000
26	Large	HANA	3	High	99.98	Up to 6	YES	9000
27	Large	HANA	4	High	99.99	Up to 9	YES	11000

Table 6.12: ERP Type CVS Data Input

```
# Store Labelled data in variable
n_ratings = len(ratings)
n_type = len(ratings['type'].unique())
n_erp = len(ratings['erp'].unique())
n_tier = len(ratings['tier'].unique())
n_criticality = len(ratings['criticality'].unique())
n_sla = len(ratings['sla'].unique())
n_s_level = len(ratings['s_level'].unique())
n_dr = len(ratings['dr'].unique())
n_price = len(ratings['price'].unique())
print(f"\n ... Content-Based Recommendation System... \n ")
print(f" Total Option : {n_ratings}")
print(f" ERP Services Type : {n_type}")
print(f" ERP Options : {n_erp}")
print(f" Number of Tier: {n_tier}")
print(f" Criticality Numbers : {n_criticality}")
print(f" SLA ratings: {n_sla}")
```

print(f" Security Levels : {n_s_level}")

print(f" Remote backup Oprion : {n_dr}")

print(f" Price Options : {n_price}")

... Content-Based Recommendation System...

Total Option	28
ERP Services Type	3
ERP Options	7
Number of Tier	4
Criticality Number	3
SLA ratings	4
Security Levels	4
Remote backup Option	2
Price Options	8

Table 6.13: Category wise Output Data

user_freq = ratings[['type', 'price']].groupby('type').count().reset_index()

user_freq.columns = ['Services', 'Options']

user_freq.head()

print(f" \n Number of Services Type : {user_freq}")

index	Services	Options
0	Large	8
1	Medium	8
2	Small	12

Table 6.14: Type of Services Options

#the above Services have a very low dataset. We will use a bayesian average

service_stats = ratings.groupby('type')[['price']].agg(['count', 'mean'])

service_stats.columns = service_stats.columns.droplevel()

print(f" \n Number of unique users: {service_stats}")

type	count	mean
Large	8	8000.0
Medium	8	3500.0
Small	12	2500.0

Table 6.15: Number of Unique Users

 # Find Lowest and Highest rated Services:

mean_rating = ratings.groupby('erp')[['price']].mean()

mean_rating.head()

print(f" \n Number of ERP Data Center Services: {mean_rating}")

erp	price
CMS	3500.0
HANA	8000.0
Miracle	2500.0

SAP	8000.0
Shree	2500.0
Tally	2500.0
UMS	3500.0

Table 6.16: Number of ERP Data Center Services

 # Lowest rated service

lowest_rated = mean_rating['price'].idxmin()

ratings.loc[ratings['tier'] == lowest_rated]

print(f" \n Lowest Price ERP Data Center Services : {lowest_rated}")

Lowest Price ERP Data Center Services : Miracle

 # Highest rated service

highest_rated = mean_rating['price'].idxmax()

ratings.loc[ratings['erp'] == highest_rated]

print(f" \n Highest Price ERP Data Center Services : {highest_rated}")

Highest Price ERP Data Center Services : HANA

References

[1] Google Colab.(June 2022).Colaboratory.https://colab.research. google.com/?utm _source =scsindex#scrollTo=rWg8C4nqrDUP

[2] Geeksforgeeks. (06 Sep, 2021). recommendation-system. https://www. geeksforgeeks.org/recommendation-system-in-python/

[3] Shaw, A. (August 2019). kaggle, https://www.kaggle.com/code/ shawamar /product-recommendation -system-for-e-commerce/notebook

[4] Verma, M. (11 Jan 2022). geeksforgeeks, https://www.geeksforgeeks.org /smart-calculator-in-python/

[5] Dwivedi, R. (18 Jan 2021). Analyticssteps. https://www.analyticssteps.com /blogs/ what-are-recommendation-systems-machine-learning

[6] Bindhu, B. (16 Oct 2019). Medium. https://medium.com/ @bindhubalu/content-based-recommender -system-4db1b3de03e7

[7] Dialani, P. (17 February 2021). analyticsinsight, https://www.analyticsinsight. net/ the-indian- government-plans-to-open-data-center-parks-in-india/

[8] Sruthi, E, (2021). Analyticsvidhya. https://www.analyticsvidhya.com blog/ 2021/06/

[9] Jeffares, A. (2019) K-means: A Complete Introduction https://to wardsdatascience. com/k-means-a-complete-introduction-1702af9cd8c

[10] Dzone. (29 may 2022). understanding-random-forest. Dzone. https:// dzone.com / articles /recommendation-engine-models

[11] Hiroshi K. (June 2011) https://jamanetwork.com/ journals/jamadermatology /fullarticle/423810