

International Conference on Machine Learning and Data Engineering

Reinforcement Learning-Based PoI Recommendation Using Reviews, Ratings and Locations Data

Ram Krishn Mishra^{a,*}, Siddhaling Urolagin^b, J Angel Arul Jothi^c

^aSchool of IT, Deakin University, GIFT City Campus, India

^bDepartment of Data Science, Middlesex University, Dubai, 111075, United Arab Emirates

^cDepartment of Computer Science, BITS Pilani, Dubai Campus, Dubai, 345055, United Arab Emirates

Abstract

The growth in internet data creates a lot of issues in decision-making for tourists to find their favorite places. To improve travel services, many internet platforms started gathering feedback from visitors in the form of numeric ratings and reviews. The numeric rating may not be a clear parameter to find the actual rating, and in such cases, a review may help in better understanding the real feedback for future travelers. The combined numeric and review data can be used to develop a better recommendation system to provide the top tourist spots based on the feedback. This research paper aims to build on a selected premise to develop a recommendation system that can suggest and map out an entire journey around a city curated in a personalized way. In order to find recommended tourist places, we use a reinforcement learning (RL) algorithm to understand the specific needs of a tourist and create a map of destinations around a selected city. The proposed model provides the suggested locations with better accuracy based on travelers' needs.

© 2026 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer review under the responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering

Keywords: Point of Interest; Sentiment Analysis; Hybrid Rating; Reinforcement Learning; Recommender System; Q-Learning

1. Introduction

Each nation is putting a greater emphasis on economic development. All nations are trying to implement seasonal laws to entice travellers from around the world and boost the number of Points of Interest (PoI) locations [7]. Over the past ten years, the tourist sector has expanded significantly and made a remarkable contribution to the world economy. Travel and food costs are incurred during tourist migration between the two locations [13]. The Internet revolution has encouraged tourism by offering real-time information about locations. It has also provided the tourism industry the numerous options for marketing and

* Ram Krishn Mishra

E-mail address: r.mishra@deakin.edu.au

developing novel strategies to attract tourists. This encouraged the individuals to continue expressing their opinions through reviews and ratings on social media platforms, blogs, and other online venues. In order to suggest recommendation systems that help consumers choose their favourite locations, this information has emerged as one of the most important resources for corporate stakeholders [15]. However, the rapid exponential expansion in data generation is caused by active online users.

Algorithms comely known as recommendation systems are very important in today's industry. Many businesses create recommendation systems to increase sales by making fresh product exhortations based on customer desire and interest. In its retail online operation, Amazon has a track record of successfully promoting various feature products [10]. After collaborative filtering emerged, recommender system research became more important for the business industry. In the literature, recommender systems have been used in several industries, including e-commerce [3], advertising [26], and tourism [17]. However, the generation of suggestions according to user preferences is complex and remains a challenge for researchers and designers. Here, systematic semantic web technologies help to resolve the recommender systems tasks efficiently.

Long-term sustainability in this sector can be improved through a hybrid recommendation system. First and foremost, it places a focus on the enjoyment, viewpoints, and experiences of visitors since they are what keep most frequent travellers coming back for more. Second, it aids organizations that specialize in destination marketing in planning advertising and marketing campaigns based on the demands of tourists.

The recommender system uses information from the different resources for predictions and suggests a particular item to the end user. Research on PoIs can benefit from the abundance of information. The fact that there are so many details available to users is still creating a serious problem. However, recommender systems are helpful in offering customized services in accordance with the requirements. To create predictions and offer users recommendations, the system compiles data from multiple sources. Travellers can identify destinations they would like to visit more easily thanks to recommendation systems, which can provide customised services based on a user's preferences. As more information is gathered online, this technology can be used in the tourism sector to spot trends and patterns in user interest, which will make it easier for tourists to find the destinations of their interest [17]. In spite of the fact that this wealth of information is useful for studying tourist destinations, it may also cause information overhead among users. Similar algorithms can be used for the tourism sector to find trends in a tourist's selection of PoIs.

The recommender systems discussed in the above paragraph are built to recommend individual PoIs to all users in a given area. This paper proposes a more curated approach to recommendation systems. The system aims to recommend an entire journey / travel route that dynamically adjusts based on the starting location of the users, the number of locations they wish to visit in a single day as well as that particular user's preference for how far they wish to travel in order to reach any given PoI. To achieve this a Q-learning system is implemented. This reinforcement learning algorithm is designed to maximize the reviews and minimize the travel distance between PoIs in a given city.

This study encompasses PoIs from cities which are highly popular tourist destination. This will improve our final system as there are more reviews from tourists along with a high number of PoIs per city allowing for a vast and broad dataset. The system was found to perform well on all cities for various starting locations. For tourists, the optimal routes often had a length of 15 – 30 PoIs to visit in order to see the best locations of the city in an optimal way.

2. Literature Survey

Several methods for the PoI recommender system have been developed to assist tourists in selecting a place. Research related to recommendation systems [1] has increased due to the open access to social networking data available on Facebook, Twitter, or Foursquare. A recommendation system is used for various applications such as online movies, friends' suggestions on social network sites, blogs, e-commerce products etc. Many researchers are trying to provide accurate recommender systems for handling domain-specific problems [18]. The integration of many main components in POI suggestion was defined and examined by researchers in [2]. It examines the role of various factors in location recommendation, including check-in frequency and social, temporal, spatial, and categorical factors. The authors also proposed two-way

models, first a ranking-based model and second a matrix factorization-based model, combining all of the major factors into a single recommendation model, including a ranking-based one. With a wide variety of experimental approaches in the machine learning area, researchers have offered a potential result for destination recommendation systems using classification techniques such as Support Vector Machine (SVM), Decision Tree(DT), and Multi-Layer Perceptions (MLP) etc. [21, 6]. Nilashi et al. [19] presented a new recommendation approach to increase predictive accuracy by applying clustering, dimensionality reduction, and prediction methods for multi-criteria collaborative filtering.

Utilizing user reviews to make predictions and recommendations is not a novel idea. In the past, numerous research studies have looked at how internet reviews affect an individual's decisions and shopping behaviours [9, 8]. In several research, the information from reviews has been used to determine the happiness and pleasure of hotel customers [28], tourist satisfaction [24], and feelings regarding a movie [22]. Online reviews are frequently described as "virtual word of mouth". These reviews are the most important information source influencing a customer's behaviour in several areas, including tourism services [16]. There is no question that these reviews have the ability to influence a user's choice and are crucial to tourism-related research and their applications. Park and Kim [20] used a cognitive fit theory to review the type and the number of reviews taken into account for the trials. In the study carried out by Tewari [25], sentiment analysis is employed to discover opinions regarding Oman's tourism from Twitter data. A part-of-speech (PoS) tagger has been used to identify entities, and these entities are compared to a domain-specific ontology. Further, Krishnamoorthy [14] develops a model to analyze internet evaluations based on five novel linguistic variables, including personal pronouns, prepositions, noun singular, noun general, and adverbs. Additionally, they have taken advantage of visibility aspects like review age and review rating. The linguistic characteristics are shown to be more accurate predictors of review usefulness.

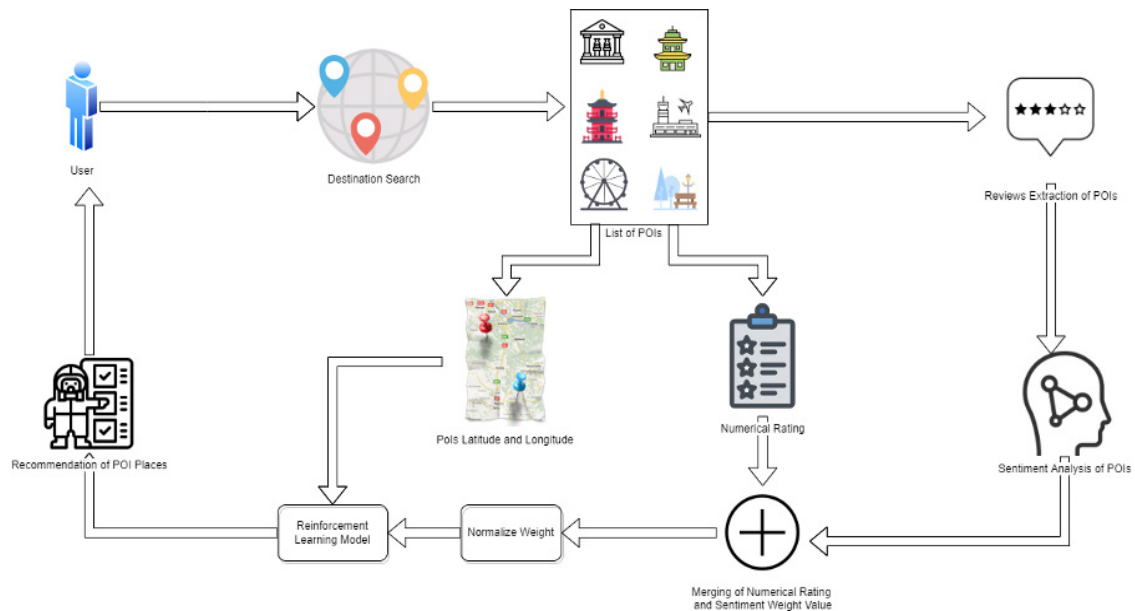


Fig. 1. POI recommender systems

Reinforcement learning [23] has been used for various problems where we aim to maximize a particular metric in an environment by taking a series of actions. Q-learning [27] is a reinforcement learning approach that can train an agent to maximize scores for future timesteps in a changing environment. Q-learning also has the added benefit of achieving optimal scores in such environments while having limited computational demands. Due to this power and flexibility, Q-learning has been used in a variety of applications [12, 11, 5] to build high-quality models across a variety of fields. The various experiments show that the cluster ensemble has better predictive accuracy than single clustering methods. Moreover, the tourism recommendation system

is presented in [4] presents the comparison between supervised and unsupervised learning. In addition, Leal et al. [15] proposed and evaluated the parallel application of profiling and recommendation algorithms in a crowd-sourced distributed database repository based on wiki.

3. Methodology

In this paper, a new PoI recommendation system is proposed. It helps the users to select their favourite places. Figure 1 shows a pictorial representation of the proposed PoI recommendation system. The rest of the work is carried out with data collection from open-source providers, followed by exploring data and required feature selection.

3.1. Dataset

We started looking for pertinent information relevant to our research in this step, with the goal of gathering reviews for PoIs. We used information from www.tourpedia.com, which offers reviews in categories like lodging, dining, points of interest, and attractions. We gathered information using web scraping from Facebook, Foursquare, Google Places, and Booking.com, four social networking platforms. These social media networks offer data on location, sentiment, reviews, ratings, and other topics. For people who are eager to travel, this information is useful for developing recommender systems. The collection includes more than 500000 locations around Europe, broken down into the following four groups: lodging, dining, points of interest, and attractions. For the following eight cities, we used points of interest (PoI) data: Amsterdam, Barcelona, Berlin, Dubai, London, Paris, Rome, and Tuscany.

3.2. Data Pre-Processing

3.2.1. Sentiment Calculation

Sentiment given by users is a text value that can proceed through the machine learning model by converting the text values to numerical values. Many tools are available to get the word numerical weight values. We used Valence Aware Dictionary and Sentiment Reasoner (VADER) in this work. VADER is a sentiment analysis model based on a set of rules to specify the polarity of text, document, paragraph, or sentence. VADER has outperformed accuracy with a small data length and can work on a few machine learning resources as it does not require much data for training. In this work, we used VADER to find the intensity of emotion on PoI reviews in the form of a numeric weight called the valence score. The valence score found for a single word will be compounded with all words in a sentence using equation 1.

$$x = \frac{X}{(\sqrt{X^2 + \alpha})} \quad (1)$$

where X is the sum of valence scores of all words, α is a normalization constant. The Table 1 shows the VADER mean compound sentiment score for 10 PoIs.

3.2.2. Numerical Rating (Polarity)

Many visitors prefer numerical ratings due to the simplicity and commonly used rating systems. The Tourpedia dataset has provided a numerical rating between 0 to 10, where 0 indicates the worst place and 10 indicates the best place. This rating system has easy comprehensibility and sensitivity, and most of the users easily give this rating due to the less time taken to fill the form and distribution of polarity is depicted in Figure 2 and Table 2 shows the mean numeric rating of 10 PoIs.

3.2.3. Merging of VADER Mean Compound Sentiment Score and Numerical Rating

In this research work, we combined the numerical rating and sentiment values by performing the addition operation because both features play an essential role in PoIs rating. Merging can help understand the view of both users who are submitting numerical and text reviews, as seen in Table 3.

Table 1. Mean sentiment value of 10 point of interest

S.No.	Point of Interests	VADER Mean Compound Sentiment Score
1	London Heathrow Airport (LHR)	2.72
2	Dubai International Airport (DXB)	2.69
3	London City Airport (LCY)	2.63
4	London Paddington Railway Station (PAD)	2.76
5	London Waterloo Railway Station (WAT)	2.77
6	Brandenburger Tor	2.76
7	London Euston Railway Station (EUS)	2.72
8	Book World By Kinokuniya	2.73
9	London Victoria Railway Station (VIC)	2.68
10	Marhaba Lounge	2.68

Table 2. Mean numerical rating value of 10 Point of interest

S.No.	Point of Interests	Mean Numerical Rating
1	London Heathrow Airport (LHR)	5.13
2	Dubai International Airport (DXB)	4.72
3	London City Airport (LCY)	6.74
4	London Paddington Railway Station (PAD)	5.33
5	London Waterloo Railway Station (WAT)	4.84
6	Brandenburger Tor	5.32
7	London Euston Railway Station (EUS)	5.51
8	Book World By Kinokuniya	6.89
9	London Victoria Railway Station (VIC)	5.16
10	Marhaba Lounge	5.59

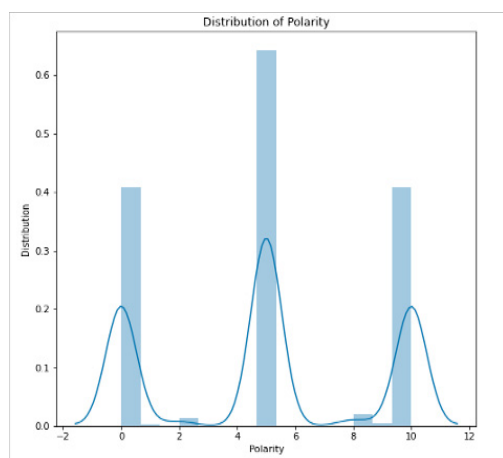


Fig. 2. Distribution of numerical rating

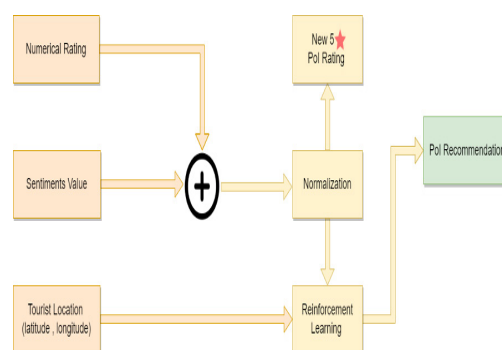


Fig. 3. Model development

Table 3. Merging of numerical rating and VADER mean compound sentiment score

S. No.	Point of Interests	VADER Score	Numerical Rating	Merged Value
1	London Heathrow Airport (LHR)	2.72	5.13	7.85
2	Dubai International Airport (DXB)	2.69	4.72	7.41
3	London City Airport (LCY)	2.63	6.74	9.37
4	London Paddington Railway Station (PAD)	2.76	5.33	7.89
5	London Waterloo Railway Station (WAT)	2.77	4.84	7.61
6	Brandenburger Tor	2.76	5.32	8.08
7	London Euston Railway Station (EUS)	2.72	5.51	8.23
8	Book World By Kinokuniya	2.73	6.89	9.62
9	London Victoria Railway Station (VIC)	2.68	5.16	7.84
10	Marhaba Lounge	2.68	5.59	8.27

Table 4. Scaled value using min-max technique

PoIs Name	Merged Values	Scaled Values
Holland Road	8	0.58
TfL Bus 17	3	0.16
Harringay Ladder	8	0.58
Stinky taxi in Dubai	13	1.00
Tramhalte J.P.Heijestraat	8	0.58
VVV Amsterdam Koffiehuis	8	0.58
TfL Bus 607	3	0.16
Hal 8 - Amsterdam RAI	8	0.58
Al Shaiba Medical Supplies	8	0.58
chequers transport ltd	8	0.58

3.3. Methods

3.3.1. Normalization

Normalization is a process to bring the numerical values to a common scale. Since sentiment values and numerical ratings have different ranges, we should get them in an equal range after merging. Normalization helps in varying scales, and if the scaling is in the acceptable content, then during the development of a model, this will help develop a good predictive model. We used min-max normalization techniques to scale the values in an acceptable range. Equation 2 represents the min-max normalization.

$$x_{norm} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (2)$$

where x is an original value, x_{norm} is a normalized value of x , x_{min} is a minimum value and x_{max} is a maximum value of dataset.

By using the min-max scaling technique, the obtained results are shown in Table 4.

3.3.2. Data for POI Location

To provide good recommendations, we must also consider the location and distance between the two POIs. This allows us to build a model that can take into account the location of POIs to plan a complete travel path rather than give individual recommendations. Specifically, we will take each POI's latitude and longitude and use that as its location, as seen in Table 5.

Table 5. Location-based POI data

POI Name	Latitude	Longitude
Langer-Franke Hauskrankenpflege	52.443178	13.383266
Dentista	43.728856	10.423021
Gekko - Tai Chi Schule	52.541762	13.324635
Ex Circoscrizione Giotto	43.460064	11.884967
Ambulatorio Pediatrico di Orentano	43.779092	10.665498

3.3.3. Model Building

The process of model development for the PoI recommender system is shown in Figure 3. The approach for developing the model is based on a combined result of a mean numerical rating and mean VADER score. The calculation of mean numerical weight is given by Equation 3. The mean value for one PoI is a summation of the numerical rating submitted by N users and getting the mean. Equation 4 explains the calculation of the mean VADER score for N users. The final PoI weight is the combination of mean numerical weight and mean VADER score which is shown in equation 5.

$$MeanNumerical_{Weight} = Mean(\sum_{U=1}^N Numerical_{Rating}) \quad (3)$$

$$MeanVADER_{Score} = Mean(\sum_{U=1}^N VADER_{Score}) \quad (4)$$

$$POI_{Weight} = MeanNumerical_{Weight} + MeanVADER_{Score} \quad (5)$$

To find the final PoI weight value, $MeanNumerical_{Weight}$ and $MeanVADER_{Score}$.

3.3.4. New PoI Ranking and Classification Algorithm

The PoI weight received from equation 5 will be used for defining the new ranking of PoI. The new ranking will have a maximum rating of 5, and the calculation process is done using equation 6. Algorithm 1 explains the classification process in the Top, Average and Normal categories using PoI Ranking.

$$PoI_{Ranking} = 5 * PoI_{Weight} \quad (6)$$

3.3.5. Reinforcement Learning Based PoI Recommendation

For this dataset, we propose Q-Learning as the reinforcement learning algorithm used to create our predictions.

a) Defining the Problem and Agent

We first define the problem in terms of Q-Learning. We will define the agent as an entity that, at any point in time, will know which POI it is currently at. This agent can then take action by deciding which POI to visit after this. We only consider POIs at one city per agent to prevent memory usage from getting too large. Therefore, our Q-Table will map from POI names to arrays where each element represents a POI. To decide its action at a given point, the agent will look up its current POI in the mapping and move to the location corresponding to the highest value in the array.

Algorithm 1 PoI Classification Based on New Rating

if $PoI_{Ranking} \geq 4$ then

$PoI_{Category} \leftarrow Top$

if $PoI_{Ranking} \geq 2$ and $PoI_{Ranking} < 4$ then

$PoI_{Category} \leftarrow Average$

else $PoI_{Category} \leftarrow Normal$

b) Evaluating Agent

The next step in Q-Learning is defining an evaluation function that will decide how well the agent performs. We naturally want to incentivize our model to go to POI with as high ratings as possible while keeping the travel required (distance to next POI) as low as possible. To incentivize high ratings, we take the rating of the POI chosen by the agent on a scale of -2.5 to 2.5 and multiply it by a constant (rating multiplier) this makes up the rating reward. To minimize distance, we take the cartesian distance between the current and chosen POI (using latitude and longitude) and multiply it by a constant (distance multiplier) this makes up the distance penalty. To get the final evaluation for the agent we subtract the distance penalty from the rating reward. This also allows us to change the values of the two multipliers and adjust how much we care about ratings and distance traveled (the constants used to produce the results shown in this paper were 1 for the rating multiplier and 0.005 for the distance multiplier). In addition to this there is also a penalty if the agent revisits the same POI in a run which we set to -2.5 to prevent the agent from getting stuck at any one place.

c) Training the Agent

We trained each agent for 30,000 episodes for each city. A single episode consists of N number of places the agent needs to visit after starting from a random location in the city. With a random probability epsilon, we decide if the agent should choose its next action using the Q-Table or if it should try a random action. We begin to decay epsilon after a fixed number of episodes to allow it to first learn by trying new ideas and then later take more of its own decisions once it has gained the necessary experience. Next, based on its actions, we get a score using our evaluation function and update the parameters of the Q-Table accordingly using Equation 7 [11].

$$Q(s, a) \leftarrow Q(s, a) + \alpha * [R_t + \gamma * \max_{a'} Q(s', a') - Q(s, a)] \quad (7)$$

$Q(s, a)$ represents the Q-value of that particular action at the current environment state. This is the value that we continue to update throughout training. Alpha(α) represents the learning rate of the model and determines how quickly it learns from new information. The current state reward (R_t) is the reward our agent has received for its current action based on our evaluation function. The discount factor (γ) controls how much the agent cares about future rewards versus current rewards. Max $Q(s', a')$ is the largest Q-value of the next timestep.

d) Hyperparameter Tuning

Hyperparameters are used to tune machine learning models for specific problems and increase their performance. Below are details for the hyperparameters of the model as well as the tuning required to achieve optimal results.

Number of Places to Visit Per Episode: we can adjust how many places the model needs to visit from a random starting location. Smaller values will encourage the model to visit the nearest locations with high ratings. Higher values for this parameter encourage the model to visit places in a path-wise manner to the

areas with the maximum POIs with high ratings even if that means a few lower rated ones on the way there. We tried a range of values such as 1, 3, 5, 10, 15, 20 as a parameter while testing.

Epsilon: This dictates how frequently the model takes random actions in hope of discovering new strategies. This value was initially set to 0.1 as it is large enough to allow the agent to quickly learn new strategies initially and was small enough to decay quickly enough to allow the agent to move forward making more and more of its own decisions.

Decay Rate: After a certain number of episodes, we decay the learning rate at a decided frequency. For these models, we started the decay after 10,000 episodes and decayed the value every 5 episodes at a rate of 0.995. We chose to decay it periodically at 5-episode intervals as it would either decay too fast or too slow for other values, which prevented it from reaching an optimal score.

Alpha: This is the learning rate of the model and determines how quickly the model learns from the information it acquires from its surroundings. We set this value to 0.1 as it provides stable training while also allowing the agent to continue learning at a reasonable pace.

Discount (must be between 0 and 1): This effects how much the agent cares about future rewards versus how much it cares about current rewards. As we highly value the future POIs the agent recommends, set this value to 0.9999.

Number of Episodes: The number of episodes we train an agent for determines how many iterations of the problem it runs through and learns from. 30,000 episodes was found to be optimal at providing the agent enough time to learn how to solve the problem for a variety of hyperparameters instead of not wasting training time with large number of episodes.

4. Experimental Results and Analysis

4.1. Experimental Setup

We used python3 libraries to perform data analysis and recommend PoIs based on the sentiment values in this work. To perform sentiment analysis and train the model, we used a Dell Precision 7920 Tower with a Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz 2.39 GHz CPU with 32GB onboard ram in a Windows 10 Pro for Workstations operating system. The relevant python3 libraries that we used include NumPy, Matplotlib and NLTK. We calculated the sentiment weight values using SentimentIntensityAnalyzer under the NTLK library and found the mean values for them. Based on the mean value of polarity and sentiment, we got rating values for each of the PoIs in our dataset.

Q-learning allows us to train agents which take decisions that can be optimized to consider both current and future rewards. This, along with the fact that we can easily alter how much certain attributes matter to us, makes Q-learning an ideal approach to this problem. Below are the values of hyperparameters used to train this algorithm.

1. Epsilon : 0.1
2. Decay Rate (gamma) : 0.995
3. Alpha : 0.1
4. Discount : 0.9999
5. Number of Episodes : 30000

The results of tweaking various environmental parameters to encourage particular behavior is shown below.

4.2. PoI Data Visualization for Analysis

a) Distribution of Numerical Rating

Figure 2 describe the numerical rating distribution of PoIs. The central tendency of the distribution lies between 4 to 6. This graph illustrates the received rating of PoIs and helps categorize the tourist spots.

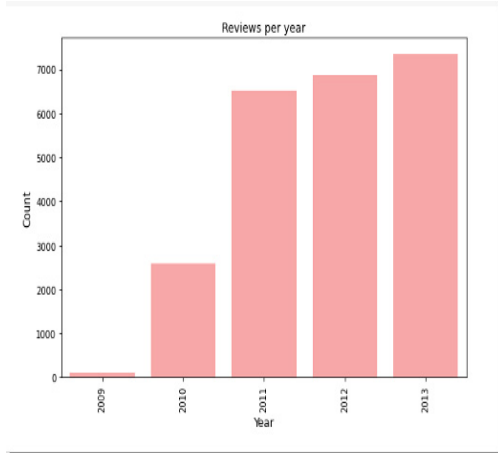


Fig. 4. Year-wise reviews

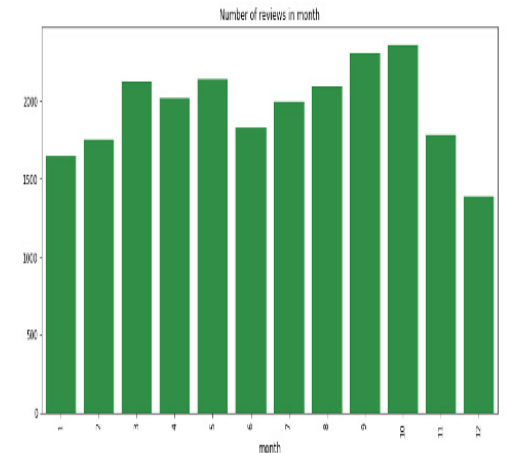


Fig. 5. Month-wise reviews

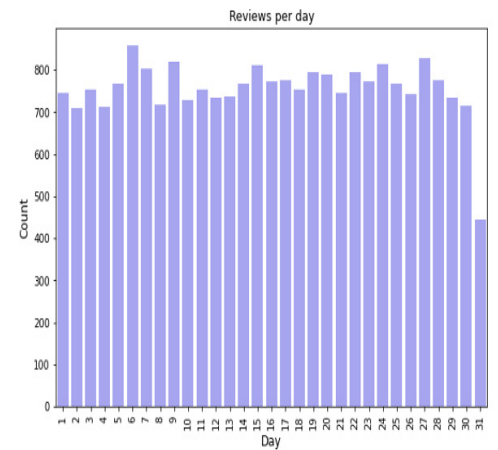


Fig. 6. Daily reviews

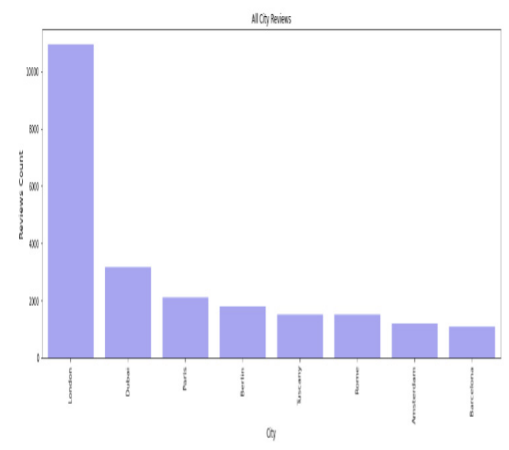


Fig. 7. All cities reviews

b) Year-wise Reviews

Year-wise reviews show the visitors' trends for the last five to ten years. This analysis shows the increase or decrease of tourist on annual basis. It finds the actual case of enhancing the services. Figure 4 describes the analysis of received polarity from the year 2009 to 2013.

c) Month-wise Reviews

Monthwise analysis gives an in-depth insight and the month of most tourists visit. Figure 5 shows that most of the visitors have given reviews in months of the October or November.

d) Day-wise Reviews Figure 6 shows the day-wise reviews counts, and most of the reviews have been received in the day of 6th, 16th, and 27th of a month.

e) All Cities Reviews Figure 7 describes the review counts which have been received for an individual city. As per the visualization, it can be observed that most of the reviews have been received for London.

4.3. Tweaking of Model Environment and Rewards

a) Number of POIs Visited Per Episode

The number of POIs visited per episodes affects how the model decides the final route. Small values (1

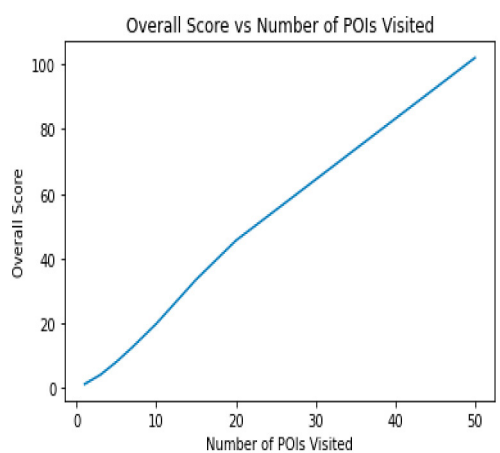


Fig. 8. Score vs number of POIs

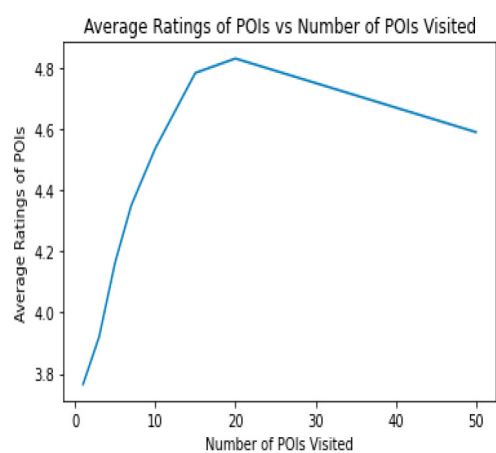


Fig. 9. Ratings vs number of POIs

– 3) of this parameter encourage the model to find fewer, high rated places in the nearby vicinity of the starting location without prioritizing the getting to locations which provide access to better POIs since it does not have enough time to visit them. Medium values (5 – 10) of this parameter prioritize finding shorter localized paths in different areas of the city that lead to the best possible POIs in that local area but cannot travel too far in order to reach POIs. Large values (15 onwards) lead the model to find the best overall travel paths throughout the city. Here the agent can sometimes go for slightly lower rated POIs initially in order to ensure that it eventually reaches the highest rated POIs almost every run. Clearly higher values lead to the agent being more consistent as it can travel to reach the best POIs when necessary. To effectively utilize this parameter, lower values must be chosen in cases where there isn't enough time to travel or visit many POIs (shorter trips). For people who plan on spending more time visiting various POIs across the city, larger values will definitely lead to the best travel routes.

Table 6 shows the scores and performance of various agents trained using a variety of values for this parameter on data collected from the city of Tuscany.

Table 6. Average scores, ratings and distance with change in number of POIs visited

SNo.	No. of POIs Per Episode	Average Overall Score over last 1000 Episodes	Average Ratings over last 1000 Episodes	Average Distance over last 1000 Episodes
1	1	1.22	3.76	-0.04
2	3	4.12	3.92	-0.04
3	5	8.09	4.16	-0.04
4	7	12.61	4.34	-0.04
5	10	19.87	4.53	-0.05
6	15	33.52	4.78	-0.05
7	20	45.60	4.83	-0.05
8	50	102.06	4.58	-0.05

Table 7. Average scores, ratings and distance with change in rating multiplier

S No.	Multiplier for Rating Reward	Average Overall Score over last 1000 Episodes	Average Ratings over last 1000 Episodes
1	0.25	9.97	4.68
2	0.5	23.25	4.92
3	0.75	35.22	4.91
3	1	45.88	4.84
4	1.5	65.31	4.70
5	2	81.53	4.83
6	3	124.63	4.90
7	5	215.66	4.67
8	7	292.04	4.66

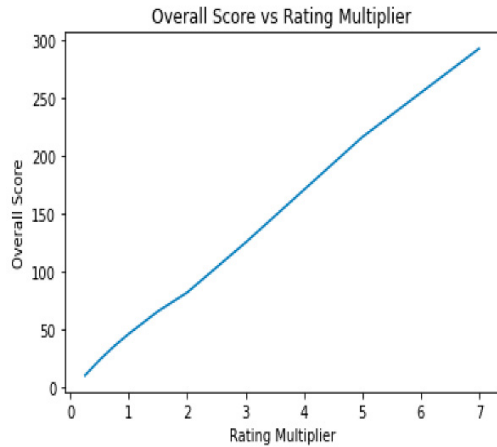


Fig. 10. Score vs rating multiplier

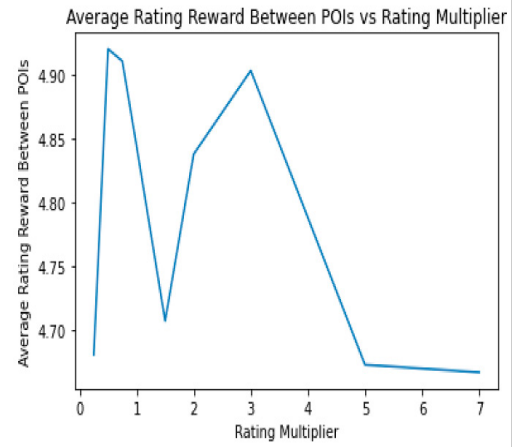


Fig. 11. Ratings vs rating multiplier

After plotting these values, we can notice certain trends, increasing the number of POIs to visit will lead to a fairly linear increase in the overall score as seen in Figure 8. Figure 9 shows us that any given city will have an optimal number of POIs to visit where the agent can lead to all the best POIs in that city. Values that are too large force it to find new POIs to visit that may not have as high ratings.

b) Multiplier for Rating Rewards

This multiplier allows us to tweak the evaluation function of the agent. During evaluation, the rating of the POI is multiplied by this constant to form the rating rewards for the agent. In theory this value should be proportional to the average rating of the POIs the agent visits. However, in practice this will also affect the overall learning of the model over time which can cause different outcomes. Table 7 shows the scores and performance of various agents trained using a variety of values for this parameter on data collected from the city of Tuscany.

After plotting these values, we can notice certain trends, Figure 10 shows that the distance multiplier has a roughly linear relationship, with higher multiplier introducing higher scores. We also notice that the rating seems to have periodic increases and decreases with an increase in the multiplier from Figure 11.

c) Multiplier for Distance Penalty This multiplier affects the agent's evaluation function by being multiplied by the Euclidian distance of the latitude and longitude between the POIs. Larger values give the agent greater penalty for large distances and encourage shorter travel times. Table 8 shows the scores and performance of various agents trained using a variety of values for this parameter on data collected from the city of Tuscany.

After plotting these values, we can notice certain trends, there seems to be an optimal distance multiplier that results in the best overall score that we can see in Figure 12. Here the value was 0.0025. Figure 13 also shows that the rating increases with an increase in the distance multiplier except in the case of 0.005 where the score goes down.

4.4. Explainability of Model Regarding Results of the Q-learning Experiments

Based on the above experiments we can select the optimal parameters for number of POIs visited, rating multiplier and distance multiplier are 20, 3 and 0.1 respectively. This agent has an overall score of 122.86, an average rating of 4.93 and an average distance penalty of -0.91.

Table 8. Average scores, ratings and distance with change in distance penalty multiplier

SNo.	Multiplier for Distance Penalty	Average Overall Score over last 1000 Episodes	Average Ratings over last 1000 Episodes
1	0.0005	45.14	4.76
2	0.001	46.68	4.84
3	0.0025	47.94	4.91
4	0.005	45.90	4.84
5	0.01	46.29	4.88
6	0.01	46.00	4.89
7	0.03	43.82	4.94
8	0.10	31.14	4.97

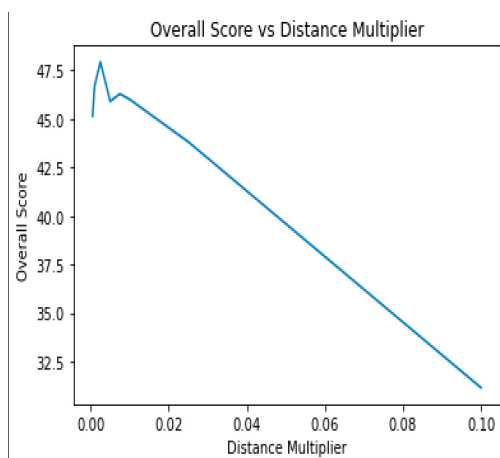


Fig. 12. Score vs distance multiplier

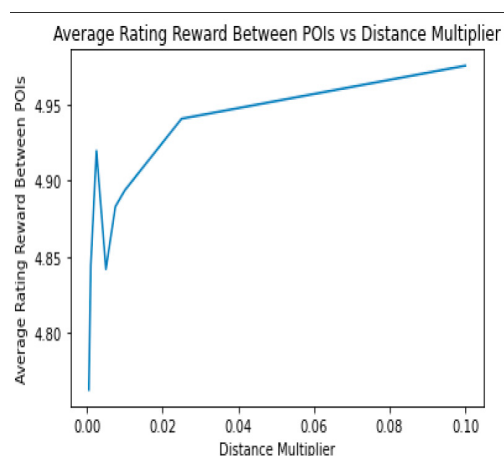


Fig. 13. Ratings vs distance multiplier

4.5. Using Q-learning Agent to Recommend POIs

To get recommendations for POIs to visit we can follow these steps.

- (i) Select a starting POI. This is the POI you are currently at or the nearest one if you are not starting at a POI.
- (ii) Use the Q-Table of the agent to get the array of learned values at that point.
- (iii) Select the index with the maximum score.
- (iv) The recommended POI is the POI in the POI list for that city, located at the index found above.
- (v) Repeat steps (i)-(iv) for as many POIs as you need to visit.

For an example case, we consider the city of Tuscany, start at the POI “Aeroporto militare Grosseto” and run the agent 20 times. This gives us the following travel path:

Aeroporto militare Grosseto → Parcheggio Scambiatore Duomo → Gate 9 → AVS Fratellanza Popolare

Peretola → Studio di Fisioterapia Dott. Alessio Pagnini → Phone& Go → ANPI - sezione di Scandicci → FLR Gate 6 → Area Di Servizio Tamoil Bar-ristorante → Parking Empoli Centro → Via delle Lenze → Treno Italo Roma Milano → Frecciarossa AV9607 Mi-RM-Na → Traghetto Toremara → ATAF - Fiesole → Moby Wonder → Freccia Rossa → Agenzia Delle Entrate → In treno → Palazzo Comunale

5. Conclusion and Future Work

In this paper, we aim to find the best route of POIs through a given city using online reviews given by previous tourists and the locations of each POI. Sentiments of user reviews played a crucial role in building this recommender system that can suggest better places for users to visit. The large volumes of tourists that travel the world allow us to create a massive dataset with reviews based of data points from numerous sources. An unsupervised Q-learning approach allowed our model to consider future POIs in order to navigate through optimal routes throughout the city. This work can be extended to allow for systems that take the users personal preferences into account based on their past travel history.

References

- [1] Artemenko, O., Pasichnyk, V., Kunanec, N., Tabachyshyn, D., 2019. Using context analysis for providing real time recommendations in e-tourism mobile location-based recommender systems. *International Scientific and Technical Conference on Computer Sciences and Information Technologies* 3, 166–169. doi:[10.1109/STC-CSIT.2019.8929822](https://doi.org/10.1109/STC-CSIT.2019.8929822).
- [2] Baral, R., Li, T., 2017. Exploiting the roles of aspects in personalized poi recommender systems. *Data Mining and Knowledge Discovery* 2017 32:2 32, 320–343. URL: <https://link.springer.com/article/10.1007/s10618-017-0537-7>, doi:[10.1007/S10618-017-0537-7](https://doi.org/10.1007/S10618-017-0537-7).
- [3] Chen, F., Huang, Y., 2019. Knowledge-enhanced neural networks for sentiment analysis of chinese reviews. *Neurocomputing* 368, 51–58. doi:[10.1016/J.NEUCOM.2019.08.054](https://doi.org/10.1016/J.NEUCOM.2019.08.054).
- [4] Dewangan, A., Chatterjee, R., 2018. Tourism recommendation using machine learning approach. *Advances in Intelligent Systems and Computing* 564, 447–458. doi:[10.1007/978-981-10-6875-1_44](https://doi.org/10.1007/978-981-10-6875-1_44).
- [5] Elavarasan, D., Vincent, P.M.D., 2020. Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications. *IEEE Access* 8, 86886–86901. doi:[10.1109/ACCESS.2020.2992480](https://doi.org/10.1109/ACCESS.2020.2992480).
- [6] Enríquez, J.G., Morales-Trujillo, L., Calle-Alonso, F., Domínguez-Mayo, F.J., Lucas-Rodríguez, J.M., 2019. Recommendation and classification systems: A systematic mapping study. *Scientific Programming* 2019. doi:[10.1155/2019/8043905](https://doi.org/10.1155/2019/8043905).
- [7] Fu, Y., Hao, J.X., Li, X.R., Hsu, C.H.C., 2018. Predictive accuracy of sentiment analytics for tourism: A metalearning perspective on chinese travel news. <https://doi.org/10.1177/0047287518772361> 58, 666–679. URL: <https://journals.sagepub.com/doi/abs/10.1177/0047287518772361?journalCode=jtrb>, doi:[10.1177/0047287518772361](https://doi.org/10.1177/0047287518772361).
- [8] Gretzel, U., Yoo, K.H., 2008. Use and impact of online travel reviews. *Faculty of Commerce - Papers (Archive)* , 35URL: <https://ro.uow.edu.au/commpapers/1190>, doi:[10.1007/978-3-211-77280-5_4](https://doi.org/10.1007/978-3-211-77280-5_4).
- [9] Gunden, N., 2017. How online reviews influence consumer restaurant selection .
- [10] Horozov, T., Narasimhan, N., Vasudevan, V., 2006. Using location for personalized poi recommendations in mobile environments. *Proceedings - 2006 International Symposium on Applications and the Internet, SAINT 2006* 2006, 124–129. doi:[10.1109/SAINT.2006.55](https://doi.org/10.1109/SAINT.2006.55).
- [11] Jang, B., Kim, I., Kim, J.W., 2019. Word2vec convolutional neural networks for classification of news articles and tweets. *PLOS ONE* 14, e0220976. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0220976>, doi:[10.1371/JOURNAL.PONE.0220976](https://doi.org/10.1371/JOURNAL.PONE.0220976).
- [12] Jeong, G., Kim, H.Y., 2019. Improving financial trading decisions using deep q-learning: Predicting the number of shares, action strategies, and transfer learning. *Expert Systems with Applications* 117, 125–138. doi:[10.1016/J.ESWA.2018.09.036](https://doi.org/10.1016/J.ESWA.2018.09.036).
- [13] Kenteris, M., Gavalas, D., Mpitziopoulos, A., 2010. A mobile tourism recommender system. *Proceedings - IEEE Symposium on Computers and Communications* , 840–845doi:[10.1109/ISCC.2010.5546758](https://doi.org/10.1109/ISCC.2010.5546758).
- [14] Krishnamoorthy, S., 2015. Linguistic features for review helpfulness prediction. *Expert Systems with Applications* 42, 3751–3759. doi:[10.1016/J.ESWA.2014.12.044](https://doi.org/10.1016/J.ESWA.2014.12.044).
- [15] Leal, F., Veloso, B.M., Malheiro, B., González-Vélez, H., Burguillo, J.C., 2019. Scalable modelling and recommendation using wiki-based crowdsourced repositories. *Electronic Commerce Research and Applications* 33, 100817. URL: <https://doi.org/10.1016/j.elerap.2018.11.004>, doi:[10.1016/j.elerap.2018.11.004](https://doi.org/10.1016/j.elerap.2018.11.004).
- [16] Lin, K.P., Shen, C.Y., Chang, T.L., Chang, T.M., 2017. A consumer review-driven recommender service for web e-commerce, in: *2017 IEEE 10th Conference on Service-Oriented Computing and Applications (SOCA)*, pp. 206–210. doi:[10.1109/SOCA.2017.35](https://doi.org/10.1109/SOCA.2017.35).
- [17] Liu, J., Jiao, X., Jin, Y., Liu, X., Liu, L., 2018. Research and implementation of poi recommendation system integrating temporal feature. *2018 IEEE 3rd International Conference on Big Data Analysis, ICBDA 2018* , 287–292doi:[10.1109/ICBDA.2018.8367694](https://doi.org/10.1109/ICBDA.2018.8367694).

- [18] Mauro, N., Ardissono, L., Cena, F., 2020. Personalized recommendation of pois to people with autism. UMAP 2020 - Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization , 163–172URL: <http://arxiv.org/abs/2004.12733><http://dx.doi.org/10.1145/3340631.3394845>, doi:10.1145/3340631.3394845.
- [19] Nilashi, M., Bagherifard, K., Rahmani, M., Rafe, V., 2017. A recommender system for tourism industry using cluster ensemble and prediction machine learning techniques. *Computers and Industrial Engineering* 109, 357–368. URL: <http://dx.doi.org/10.1016/j.cie.2017.05.016>, doi:10.1016/j.cie.2017.05.016.
- [20] Park, D.H., Kim, S., 2008. The effects of consumer knowledge on message processing of electronic word-of-mouth via online consumer reviews. *Electronic Commerce Research and Applications* 7, 399–410. doi:10.1016/J.ELERAP.2007.12.001.
- [21] Portugal, I., Alencar, P., Cowan, D., 2018. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications* 97, 205–227. URL: <https://doi.org/10.1016/j.eswa.2017.12.020>, doi:10.1016/j.eswa.2017.12.020.
- [22] Sahu, T.P., Ahuja, S., 2016. Sentiment analysis of movie reviews: A study on feature selection and classification algorithms. *International Conference on Microelectronics, Computing and Communication, MicroCom 2016* doi:10.1109/MICROCOM.2016.7522583.
- [23] Sutton, R.S., 1992. Introduction: The challenge of reinforcement learning. *Reinforcement Learning* , 1–3URL: https://link.springer.com/chapter/10.1007/978-1-4615-3618-5_1, doi:10.1007/978-1-4615-3618-5_1.
- [24] Taecharungroj, V., Mathayomchan, B., 2019. Analysing tripadvisor reviews of tourist attractions in phuket, thailand. *Tourism Management* 75, 550–568.
- [25] Tewari, V., 2019. Seasonality in tourism: The case of oman. *Ottoman Journal of Tourism and Management Research* 4, 463–476. doi:10.26465/OJTMR.2018339522.
- [26] Vani, M.S., 2016. A recommender system for online advertising. *International Journal of Advanced Research in Computer and Communication Engineering* 5, 599–604. doi:10.17148/IJARCCCE.2016.52137.
- [27] Watkins, C.J.C.H., Dayan, P., 1992. Q-learning. *Machine Learning* 1992 8:3 8, 279–292. URL: <https://link.springer.com/article/10.1007/BF00992698>, doi:10.1007/BF00992698.
- [28] Zhang, Y., Vásquez, C., 2014. Hotels' responses to online reviews: Managing consumer dissatisfaction. *Discourse, Context Media* 6, 54–64.