

Article

A Map-Based Recommendation System and House Price Prediction Model for Real Estate

Maryam Mubarak¹, Ali Tahir^{1,*}, Fizza Waqar², Ibraheem Haneef³, Gavin McArdle⁴, Michela Bertolotto⁴ and Muhammad Tariq Saeed⁵

¹ Institute of Geographical Information Systems, National University of Science & Technology, Islamabad 44000, Pakistan; mmubarak.ms18igis@gis.nust.edu.pk

² GIS Plus Total Solutions, Islamabad 44000, Pakistan; fizza.waqar@gisplus.net

³ Department of Mech & Aerospace Engg, Air University, Islamabad 44000, Pakistan; ibraheem.haneef@mail.au.edu.pk

⁴ School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland; gavin.mcardle@ucd.ie (G.M.); michela.bertolotto@ucd.ie (M.B.)

⁵ Research Centre for Modelling & Simulation, National University of Science & Technology, Islamabad 44000, Pakistan; tariq@rcms.nust.edu.pk

* Correspondence: ali.tahir@gis.nust.edu.pk



Citation: Mubarak, M.; Tahir, A.; Waqar, F.; Haneef, I.; McArdle, G.; Bertolotto, M.; Saeed, M.T.A. Map-Based Recommendation System and House Price Prediction Model for Real Estate. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 178. <https://doi.org/10.3390/ijgi11030178>

Academic Editor: Wolfgang Kainz

Received: 3 January 2022

Accepted: 3 March 2022

Published: 7 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Simple Summary: The accessibility of spatial big data help real estate investors to make better judgement calls and earn additional profit. Since location is considered necessary for real estate and consequent decision-making, digital maps have become a prime resource for real estate purchases, planning and development. Personalisation can support in making judgments by identifying user requirements and inclinations, which a user interacts with digital map, it records all the user's activities. A personalised real estate portal can use this information to suggest properties, assist homeowners and provide valuable real estate analytics. By monitoring user interactions through an online real estate portal, the framework provided in this article can make personalised recommendations of real estate based on content, collaboration and location. The effectiveness of the recommendations was tested by the user feedback mechanism through a method of mean absolute precision, and the results show that 79% precise suggestions were generated. Out of 5 recommendations produced, users were interested in at least 3. A separate house price prediction model was also developed base on neural networks and classical regression technique. This model implemented to assist users in making an informed decision regarding prospects of real estate purchase.

Abstract: In 2015, global real estate was worth \$217 trillion, which is approximately 2.7 times the global GDP; it also accounts for roughly 60% of all conventional global resources, making it one of the key factors behind any country's economic growth and stability. The accessibility of spatial big data will help real estate investors make better judgement calls and earn additional profit. Since location is deemed necessary for real estate and consequent decision-making, digital maps have become a prime resource for real estate purchases, planning and development. Personalisation can assist in making judgments by identifying user desires and inclinations, which can then be recorded or captured as a user performs some interactions with a digital map. A personalised real estate portal can use this information to suggest properties, assist homeowners and provide valuable real estate analytics. This article presents a novel framework for recommending real estate to users. By monitoring user interactions through an online real estate portal, the framework can make personalised recommendations of real estate based on content, collaboration and location. The effectiveness of the recommendations was tested by the user feedback mechanism through a method of mean absolute precision, and the results show that 79% precise suggestions were generated, i.e., out of 5 recommendations produced, users were interested in at least 3. Along with that, a separate house price prediction model based on neural networks and classical regression techniques was also implemented to assist users in making an informed decision regarding prospects of real estate purchase.

Keywords: real estate; map personalisation; map recommendation; house price prediction; estatech maps; real estate analytics

1. Introduction

Driven by advertising technologies and goals to produce targeted ads, the personalisation and customisation of websites and services have become the new norm in our society. The need for personalisation has been driven by the increase in data and information available. Information overload, which makes it challenging to find relevant information, has been a phenomenon for the past two decades. For example, a study from 2003 found that unique information creation was estimated to be between 1 to 2 exabytes. This implied that each human being must be processing 250 megabytes of information. Almost 20 years later, this demonstrates the mounting need for efficient and accurate user recommendation systems to help find pertinent data and information. Personalised content delivery to any set of users may consist of multiple aspects.

A factor that plays a vital role in most personalised web interfaces is the interactivity and the “user-friendly” nature of the User Interface (UI). Every web user, be it a novice, or an expert, wants the interface to provide meaningful content delivered without having much prior expertise about its functionality. This process involves a lot of work from a web developer’s perspective but should be invisible and seamless to the end-user. Therefore, various tools and techniques have been developed to implicitly collect data from users. Implicit data collection, in simpler terms, is just the collection of a user’s data through “interface interactions” without the user having to provide the data in a specific manner. The data is then used to determine interests and make recommendations. At the same time, another aspect growing in popularity is having the location information of a user to make recommendations.

Such recommender systems are widely deployed in many consumer domains, such as online shopping, although our research focuses on real estate recommendations. Real estate recommendation is often about the location of a property item, so we have incorporated online map interactions as a tool to understand a user’s interests. This paper presents four principle recommendation approaches for effectively identifying property items in our real estate portal. (1) Analysis and implementation of content-based filtering for suggesting real estate items. (2) Collaborative filtering approach reduces the computational cost by suggesting similar items to a similar group of users. (3) Location-based approach for predicting the area of interest to the user based on geographical location and user preferences. (4) Building a price prediction model to assist users in making an informed decision. The reason for selecting the first two approaches is based on the fact that the features of a real estate database closely resemble a movie database. Both content-based filtering and collaborative filtering have proven to provide precise recommendations to users [1]. Introducing a location-based approach is essential since property items have an inherent location aspect.

We have used data from the Estatech map’s portal: <https://www.the-estatech.com> (accessed on 15 September 2021) for the recommendation part of the study. We also obtained data in explicit and implicit formats. In addition, historical data of properties and price listings were obtained from [Zameen.com](https://www.zameen.com) (accessed on 15 September 2021), a real estate portal for online property listings. The techniques and methods used for recommendation algorithms were the score tree processes, TF-IDF and K-nearest neighbours. For house prediction, we cross-compared two techniques, namely multiple linear regression and Keras regression based on neural networks.

The remainder of the article is organised as follows: Section 2 presents the related literature review. The methodological approach is given in Section 3. Section 4 presents a discussion and the results, while Section 5 concludes the study and provides future recommendations.

2. Related Work

Today's modern recommendation engines have emerged from the domain of information filtering, a term created by [2] outlines one solution for the issue of retrieving the correct information against a pool of massive online data, called content filters. To ascertain a user's choice correctly, multiple visualisation tools have also been developed to accurately distinguish a user's interests and inclinations. These tools can also be considered as a form of content filter. This domain has been progressing ever since. [3] demonstrate various options for integrating a recommendation engine into a real estate portal's user journey. Furthermore, in the same manner, the work validated how additional real estate details can provide more accurate recommendation results when integrated into the proposed model of deep learning and factorisation machines.

Another study by [4] aims to determine if consumer loyalty will help a recommender system be more accurate. Other techniques implemented by [5] such as using intelligent data analysis methods to create a recommender framework to solve the problem of recommending the most appropriate components for each user at any given time. They have further addressed the problem of converting an original dataset from a real component-based application to an optimised dataset. After gathering the interaction data and developing a dataset to produce optimised recommendation results, machine learning algorithms using feature engineering techniques and feature selection methods were also applied. Users and developers alike want information processing and its display to be swift. The system developed by [6] is based on an implicit profiling system for tracking the user's interests through mouse movements.

A gap analysis approach by [7] identifies the differences between theory and reality in presenting information on location choice by developing a seven-factor classification tool for evaluating property websites. To capture the relations between the latent feature vectors of real estate items, Ref. [8] utilised the average-based and individual-based geographical regularisation terms. Both terms are integrated with the weighted regularised matrix factorisation framework to model users' implicit feedback behaviours to provide them with personalised property recommendations.

A probabilistic model for collaborative filtering by [9] calculates the predicted values for items against active users, given that there is information already available about those active users. The same research divides collaborative filtering methods into two primary modules, memory-based collaborative filtering and model-based collaborative filtering. Additional probabilistic approaches have been presented, some more sophisticated than others, including the work of [10]. The recommended procedure is taken as a sequential decision-making process, and the use of Markov decision chains have been suggested to create a model. However, they do not state any improved accuracy over Breese's projected models. Another recommendation system by [11] applies content-based filtering, a fuzzy technique for identifying similar and different content and a prediction algorithm for identifying the right set of movie content for the user. At the same time, Ref. [12] developed item to item centred algorithms. It has been done to provide improved outcomes than user-based algorithms by comparing the approach with K-nearest neighbour.

In the domain of GIS, a complete map personalisation system is developed by [13] in which the users' interests are implicitly recorded and given specific rankings based on certain criteria fulfilment upon user's mouse clicks or movements. As already mentioned, map personalisation has become an area of interest since data overload has become a common scenario in spatial information systems. In the model developed by [14], the entire focus is to understand map usage patterns of the end-users. The goal is again focused on developing personalised maps for users on a web interface. Working on similar lines, RecoMap [13], is a web-based platform through which each user receives customised spatial recommendations based on their likings. The results are presented in a map interface highlighting the user's personalised spatial recommendations. The adaptive map also shows the user's preferences and the context in which they are used. A different approach by [15] is to build a recommendation system and map interface, represented in a

personalised format for the user to acquire quick results. Further inferences are made by studying the user's behaviour for system improvement.

Another recommender system designed by [16] is for real estate users who do not have a user profile for any real estate portal. The session-based interaction of the user is made more effective by utilising a user's search context and ranking criteria for any suitable property item. A portal developed by [8] specifically designed for real estate uses two basic approaches for user profiling, an ontological structure and case-based reasoning. The purpose is to save the end-user from the stress of massive online searching and deliver results where the user gets quick recommendations based on their interests. A recommendation system that is being used by the US-based real estate website "Trulia" utilises a "square counting method" [17]. The method works well with large scale datasets and delivers swift results per the user's preferences based on love and hate edge configurations.

Things have changed significantly in the real estate industry during the COVID-19 era. In some regions, house prices have shown signs of stagnancy and even, in some cases, decreasing trends as people lost their livelihoods. These conditions have urged people to tread more carefully while making investments in this sector. In such a scenario, a price prediction model can help users make an informed decision. A method by [18] for predicting house prices utilises a Mallows model averaging estimator, which is vigorous in terms of spatial dependence. Another study on ML models for house price prediction by concludes that the random forest regressor model provides the best results amongst all other compared models like linear regression, decision tree, k-means regression [19]. Another similar study carried out by [20] applies regression as a predictive model. They use MSE, MAE and RMSE as their evaluation metrics for their model's accuracy. Another interesting study by [21] used Multiple Regression Analysis (MRA) to estimate property prices for mass evaluation. The structural qualities and the property's location were viewed as two primary micro factors of house pricing. MRA was utilised to determine the structural characteristics and locational attributes that statistically influence house price using a sample of 106 house sale transactions from 2011 to 2015. An alternative approach by [22] focuses on traditional solutions based on widely known methods and procedures and faith in the infallibility and objectivity of a human analysing the real estate market. Since modern technologies are also boldly entering the arena. Hence, the study's key focus is that organisations should stop viewing automated solutions (such as AVM, CAMA, and AAVM) as functioning in opposition to traditional approaches and instead embrace them as supplemental tools.

Our previous work in map personalisation discusses the initial concept of personalisation using real estate analytics [23]. It also evaluates background research relating to the building blocks that lead to a recommendation engine for real-time analytics. Extensive research in this field has revealed gaps between real-estate analytics and map-based personalisation, recommendation and prediction; thus, we have tried to bridge this gap in our research and initial development work. We also found motivation for our study and consequent development since map-based personalised real estate portals do not widely exist in the online real estate market. Having to sift through a plethora of online data is no longer suitable for most users, and personalisation has become a key concept in every aspect of data search. In our scenario, real estate test users have been interacting with a real estate portal, "Estatech Maps", to search and post property items. Our recommendation system is based on three techniques. This includes content, collaboration and location-based filtering. The interaction of users is captured via the map-based interface of the real estate application, Estatech Maps, and stored in a database. Based on this data and analysis, a user gets recommendations as per their area of interest. Along with that, we have incorporated a module based on traditional regression techniques and Keras API for predicting the future price trends of property items.

The subsequent section discusses the detailed insight of the research process regarding data collection, its pre-processing, run time environment creation, and model conception. Finally, the section will discuss the following crucial areas of the research process in

detail. (1) Data collection and Technology. (2) Property Recommendation. (3) Price prediction model.

3. Methodology

“Estatech Maps” main focus is to provide personalised real estate listings to its users on a map-based interface by making accurate recommendations and providing insight about price trends of a user’s area of interest. Recommendation and price prediction were the key focus areas to deliver map-based personalisation to the users. In the first stage, a detailed study on the mathematical interpretation of recommendation algorithms was carried out. The second stage focused on the algorithm’s designs, and in the third stage, development based on those algorithms was carried out, and the models were implemented. The validation and testing of these models were carried out in the final stage of the research. The sequence of the study is illustrated in Figure 1.

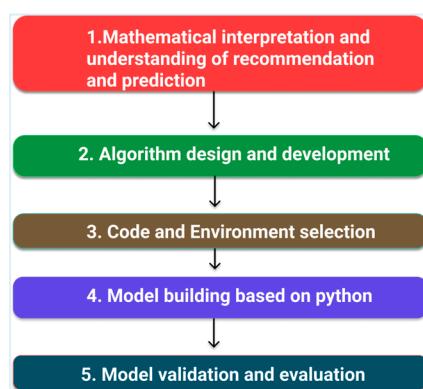


Figure 1. Methodology Sequence.

Regarding price prediction, after researching various prediction techniques, two models were selected. One is based on a classical regression technique, and the other relies on neural networks.

3.1. Data Collection and Technology

User interaction data was extracted from the portal over a year (May 2020–March 2021). Data were extracted in JSON format from a MongoDB database, which was converted to a CSV format. It consisted of 1600 recorded user interactions with the portal. The data for house price prediction was acquired from a Pakistani based real estate portal [Zameen.com](#) (accessed on 15 September 2021) for two years between 2019–2020 for Islamabad City.

Both the datasets from Estatech maps and [Zameen.com](#) (accessed on 15 September 2021) were converted into test and training datasets. [Zameen.com](#) (accessed on 15 September 2021) data, used for a house price prediction model, was further converted into a validation dataset. The data consisted of multiple files: User login information (User demographics), Interaction Data (Most viewed properties list) and Item Data (Properties).

TuriCreate was used to build the recommendation engine for content-based and collaborative filtering, whereas a K-means clustering technique was employed for the location-based recommendation. TuriCreate is an open-source toolkit for building Core ML models for tasks like image recognition, object detection, style transfers, and recommendation generation, among others.

Tensor Flow and Keras API were used as baseline technologies to build the house price prediction model and a proper validation for model loss and model accuracy, which was done through evaluation techniques of MSE, MAE and RMSE. TensorFlow is a machine learning software library that is free and open-source. It can be used for various activities, but it focuses on deep neural network training and inference. The Google Brain team created TensorFlow for internal Google use. In 2015, it was published under the Apache License 2.0.

The reason for using TensorFlow is that it is an open-source artificial intelligence library that builds models using data flow graphs. It enables programmers to create large-scale neural networks with multiple layers. Keras is a deep learning API written in Python that runs on top of the TensorFlow machine learning system. It was built with the objective of allowing fast experimentation.

3.2. Property Recommendation

The three areas of focus for the recommendation engine are discussed in detail in each of the following sections.

3.2.1. Content-Based Filtering

The concept behind recommender systems is data analytics. This can be achieved either by score-based algorithms or by suggesting to a user the top items in an N-th list of item array. In our scenario, our recommender system is designed for suggesting property items listed for sale or rent. If a person has interacted with a map-based interface with a property item, say in area “A” with attribute array “X”. The recommender system can display similar items for the user in an instant and accurate manner.

In content-based filtering, the angle between the user’s profile and the items the user is interested in is determined. This cosine angle determines how close in space the vectors lie to each other and is also termed cosine similarity. The closer they are, the more similar they are deemed. Let us consider a vector “U” of users {user1, user2, user3 . . .} and a vector “P” of property items {p1, p2, p3, p4 . . .}. The similarity between these two vectors can be calculated as:

$$\text{sim}(U, P) = \cos(\theta) = \frac{U \cdot P}{\|U\| \cdot \|P\|} \quad (1)$$

In other words:

$$\text{sim}(U, P) = \frac{\text{number of people who viewed both P1 and P2}}{\text{number of people who viewed either P1 or P2}} \quad (2)$$

The cosine value or similarity in Equation (1) can range between -1 and 1 . Based on this value, the articles are organised in descending order, and the top recommendations are made to the user.

The approach for content-based filtering is further explained in Figure 2, which shows how a tree-based criterion for item selection works. The concept is based on how much interactivity a user has with a specific item or category. Interest ratios are calculated between corresponding categories based on “incrementing the value of frequency”. For example, buyers’ interactions with rent or purchase categories define the interest ratio between the two categories. The flow of the function which performs frequency calculation is elaborated in Figure 3, which details another content-based filtering process, namely *TF-IDF*. For example, suppose a user searches for “the rise of analytics” on Google. In that case, it is inevitable that the word “the” will occur more frequently than “analytics”, but the relative importance of analytics is higher than the search query point of view. In such cases, *TF-IDF* weighting negates the effect of high-frequency words in determining the significance of an item (document).

$$TF(t) = \frac{\text{Frequency of term } "t" \text{ in document}}{\text{Total number of terms in document}} \quad (3)$$

$$IDF(t) = \frac{\log_{10}(\text{Total Number of documents})}{\text{Number of documents containing } "t"} \quad (4)$$

$TF(t)$ is simply the frequency of a word in a document, whereas $IDF(t)$ signifies the rarity of the word, so if the word occurring in the document is less, then the value of IDF increases. In Equation (4), the log parameter is used to dampen the effect of high-frequency words. We have utilised both the score tree process and *TF-IDF* approaches in formulating

our content-based filtering algorithm. Initially, user-user similarity and item-item similarity are obtained in an array format. The next step in the process was the creation of the item-user similarity matrix.

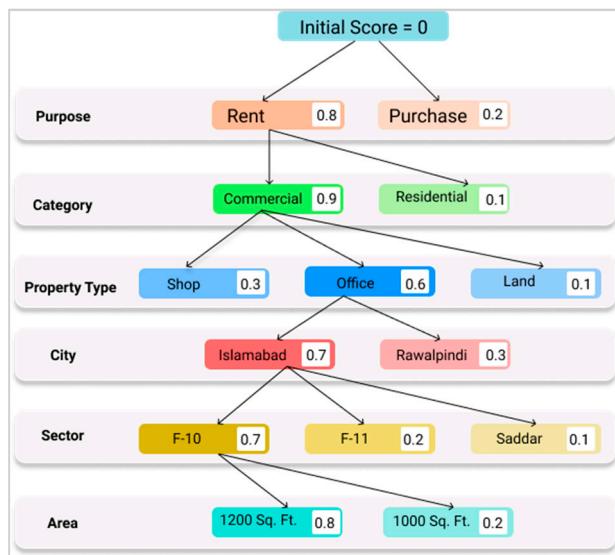


Figure 2. Score Tree process for Property Selection.

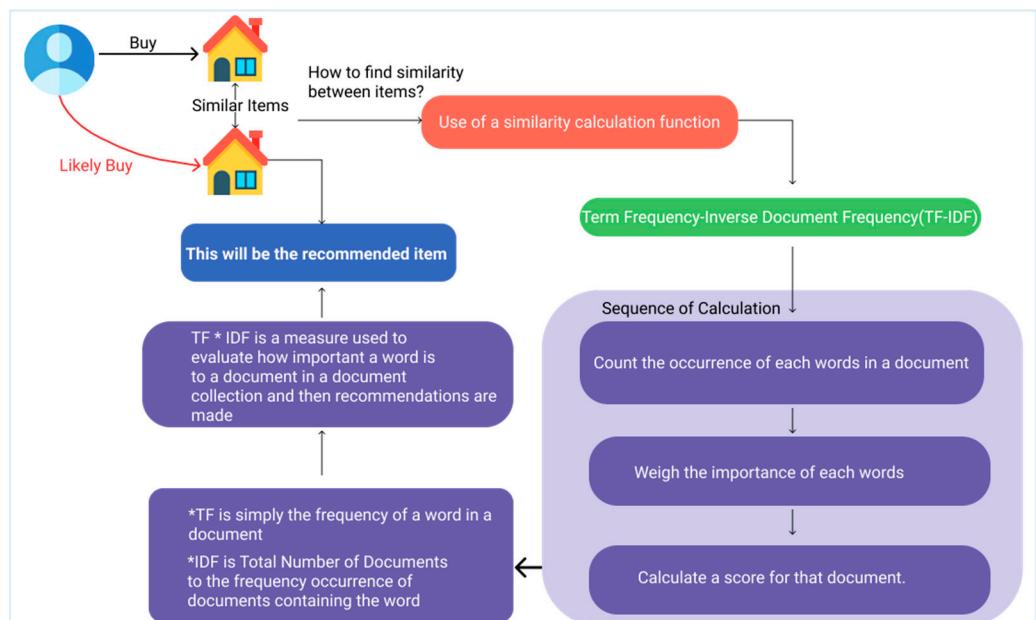


Figure 3. Frequency Calculating Function for Content-Based Filtering.

3.2.2. Collaborative Filtering Approach

In our approach towards developing a collaborative filter for the portal, the test users were divided into segments based on their preferences and items were recommended as per mutual choices of users belonging to that segment. The more the user interacts with items on display and rates them, the more precisely the system can suggest appropriate items. The algorithms designed for collaborative filtering are mostly based on finding the similarities between users on the grounds of the rank or rating they have given to previous items. So, for predicting any item for user " u ", calculations are made to compute the weighted sum of user " u " given by users to an item " i ". The prediction $PD_{u,i}$ would then be calculated as:

$PD_{u,i}$ is the prediction term for user ' u ' against an item " i ".

$$PD_{u,i} = \frac{\sum_v (i_{v,i} * s_{u,v})}{\sum_v s_{u,v}} \quad (5)$$

$PD_{u,i}$ is the prediction term for user ' u ' against an item " i ", $i_{v,i}$ is the interaction by the user say " v ". with an item " i ", $s_{u,v}$ is the likeness among the two users, i.e., user " u ". and user " v ".

As per Table 1, the interactions between users and properties is recorded, and suggestions to a new user "U1" are generated. At the same time, the symbol "x" represents any interaction between a user and a property item. It is evident that there is more similarity between user 1 and user 2 than user 3. Based on this, user 1 and user 2 will be grouped together for future recommendations. Algorithm 1 depicts a generalised algorithm that has been designed for grouping user 1 and user 2 together so that the same properties get recommended to them.

Table 1. Property Interaction Matrix.

	P1	P2	P3	P4
U1	x	x	-	-
U2	x	x	x	-
U3	-	-	-	x

Algorithm 1 Collaborative Recommendation Algorithm for New User "U1"

- 1: Input: Properties Dataset → all properties
- 2: Neighbours used for ranking → K
- 3: New User for recommendation → U1
- 4: Current recommendations for New User U1 → \emptyset
- 5: Users location history → L
- 6: rank = 0
- 7: **Output:** N items to be recommended
- 8: **For each** → property \in all properties **do**
- 9: if (users for P1==users for P2) **then**
- 10: rank++
- 11: Group according to the nearest neighbour in similarity (K, property, user, L) = users for P1&&user for P2
- 12: Recommendations [U1] → [P3]
- 13: **Descending rank. sort** (properties)
- 14: **Return** Recommendations []

3.2.3. Location-Based Filtering

The purpose of a location-based recommendation system is to recommend items based on the geographical location of a user. In this case scenario, recommendations can also be made possible for a new user (cold start problem) where items get recommended based on users of nearby locations who may align with the new user based on other parameters such as age or gender etc. A location-based recommendation can immensely benefit people in saving time and travel costs when displayed effectively through an interactive interface.

Equation (4) calculates the probability of interactivity of a user with an item " i " established based on distance from all previous interactions of the user, which, in our case, are other property items. Whereas in Algorithm 2 the algorithm for test user 1 has been specified.

$$L_{geo}(u, i) = \prod_{k \in I} f(\text{distance}(i, k)) \quad (6)$$

In Algorithm 2, a generalised algorithm for calculating location-based recommendations for users is presented. It considers at least 50 users in a cluster for a similarity score calculation.

Algorithm 2 Location-based recommendation algorithm for New User “U1”

- 1: Input: A user
 - 2: Collection of users → U
 - 3: Users location history → L
 - 4: Similarity matrix between users → M
 - 5: Current recommendations for New User based on location → ∅
 - 6: Count = 0
 - 7: Output: Top N location-based property recommendations based on users' similarities and preferences
 - 8: M = similarity matrix values
 - 9: Number of nearby users selected for similarity
 - 10: score calculation ≤ 50
 - 11: For each → user ∈ U do
 - 12: LOC = location discovery // level of hierarchy or granularity of location
 - 13: Calculate similarity distance score
 - 14: Calculate distance from nearby users
 - 15: The similarity score of User U1's last x interacted properties == similarity score of nearby user's similarity score
 - 16: Sort properties based on a count
 - 17: Select top N scores
 - 18: Select top N properties
 - 19: Return N Recommendations
-

3.3. Price Prediction Model

The critical aspect to notice in the price prediction model is that the data used for this analysis is the “offered set of prices” by the real estate portal [Zameen.com](#) (accessed on 15 September 2021). These prices can change as per the market variations or any redundancy in the real estate sector.

For the prediction and analysis aspect, two regression techniques, namely (1) Multiple linear regression and (2) Keras regression, were selected. The cross-comparison and validation of these techniques were performed. The one that performed better in terms of variance score was selected as the final model for visualising house prices.

3.3.1. Multiple Linear Regression

This is a type of linear regression in which the supposition is that the independent variable y and the dependent variable x have a linear or direct relationship. We used the Sklearn library to import the Linear Regression module. As already mentioned, our dataset was divided into a test set and a train set.

3.3.2. Keras Regression

We use regression techniques to predict the independent variable y , which is price. We have 14 features (property_id, location_id, property_type, price in pkr, price in dollars, location, city, province, bedrooms, bathrooms, area purpose, date of addition to the portal, area in Marla, area in sq. ft,); therefore we selected 14 neurons as baseline along with one output and one input layer for the model. There are 4 hidden layers.

The model was trained for 400 epochs, with the training and validation precision being recorded during each cycle. Finally, the model was run on both train and test results, with the loss function being measured at each epoch to keep track of how well the model is performing.

4. Results and Discussion

4.1. Content-Based and Collaborative Filtering Model Building

We adopted the Sklearn library as it contains a module called pairwise distance, which identifies any two items which have similar characteristics or any two users who have similar interests. To apply such a distance, we defined a function that returns the parameters of interactions, similarity, and the type against which we are obtaining the similarity. The algorithm generates suggestions based on the user's profile (collaborative filtering model) for the first case. For the second case, the suggestions are based on the item's attributes (content-based filtering). In the end, we were able to obtain recommendations for both users and items. In Tables 2 and 3, it can be seen that for all users, "U", scores "S" are obtained in descending order, with the highest similarity scores at the top.

Table 2. Manual Model, Collaborative Filtering Scores.

	S1	S2	S3		S(N-2)	S(N-1)	S(N)
U1	2.065	0.734	0.629	...	0.393	0.393	0.392
U2	1.763	0.384	0.196	...	-0.088	-0.086	-0.086
U3	1.795	0.329	0.158	...	-0.136	-0.134	-0.134
U4	1.591	0.275	0.102	...	-0.167	-0.166	-0.166
U5	1.810	0.404	0.275	...	-0.009	-0.008	-0.008

Table 3. Manual Model, Content-Based Filtering.

	S1	S2	S3		S(N-2)	S(N-1)	S(N)
U1	0.446	0.475	0.505	...	0.588	0.573	0.566
U2	0.108	0.132	0.125	...	0.134	0.136	0.137
U3	0.085	0.091	0.087	...	0.084	0.089	0.090
U4	0.032	0.045	0.042	...	0.053	0.051	0.052
U5	0.157	0.174	0.189	...	0.199	0.197	0.200

In Tables 4 and 5, it is observed that the scores obtained against each user are not easy to interpret. It is not clear against which property ID the user is getting the suggested items of interest. To make our results clearer, we have utilised the Turicreate library. This made the results obtained easier to understand. Table 4 represents content-based recommendation model results. The model was assessed for five users of the portal, and recommendations were generated for them.

In Table 4 the set of 5 users are recommended the same 5 property items due to the popularity of those items as being the most interacted with.

Table 5 shows the properties recommended to users based on grouping with other users having similar interests. Property IDs having a higher score are ranked higher. Each user is recommended a different set of properties, which clearly shows that personalisation exists for each user.

Table 4. Content-Based Recommendation using Turicreate.

User ID	Property ID	Interaction	Rank
1	1599	9.0	1
1	1201	7.0	2
1	1189	5.0	3
1	1122	4.0	4
1	814	3.0	5
2	1599	9.0	1
2	1201	7.0	2
2	1189	5.0	3
2	1122	4.0	4
2	814	3.0	5
3	1599	9.0	1
3	1201	7.0	2
3	1189	5.0	3
3	1122	4.0	4
3	814	3.0	5
4	1599	9.0	1
4	1201	7.0	2
4	1189	5.0	3
4	1122	4.0	4
4	814	3.0	5
5	1599	9.0	1
5	1201	7.0	2
5	1189	5.0	3
5	1122	4.0	4
5	814	3.0	5

Table 5. Collaborative Filtering Using Turicreate.

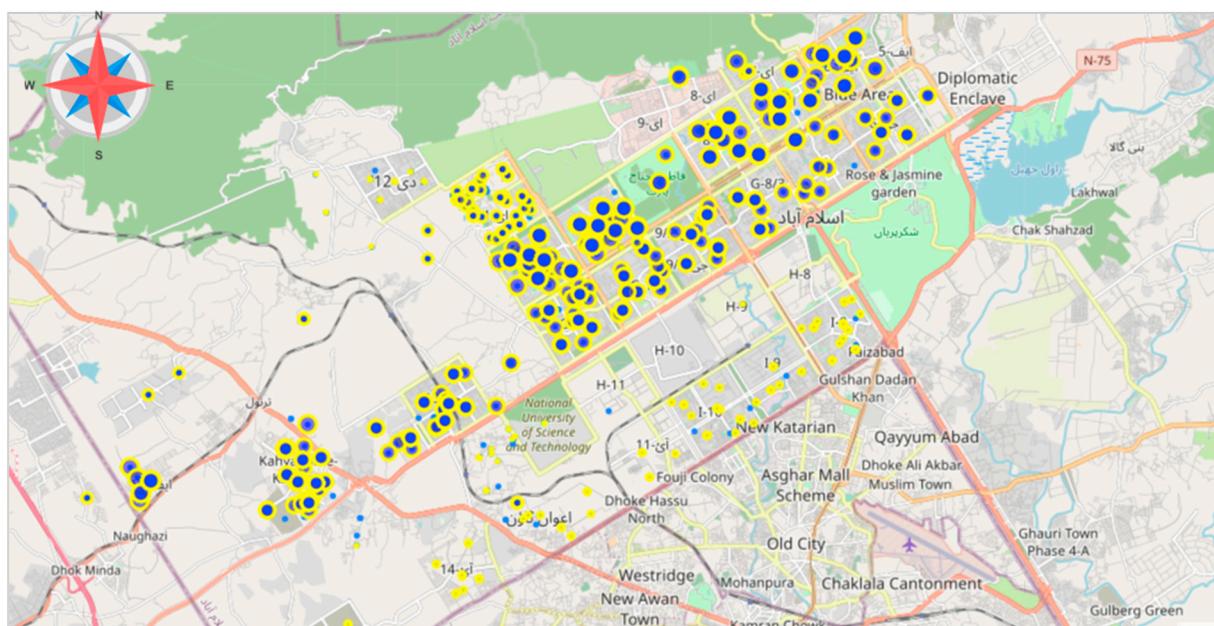
User ID	Property ID	SCORE	Rank
1	327	0.989	1
1	409	0.953	2
1	599	0.814	3
1	487	0.783	4
1	551	0.759	5
2	50	1.116	1
2	171	1.075	2
2	431	0.923	3
2	005	0.832	4
2	137	0.793	5
3	333	0.626	1
3	388	0.603	2
3	375	0.544	3

Table 5. Cont.

User ID	Property ID	SCORE	Rank
3	381	0.536	4
3	392	0.522	5
4	055	1.113	1
4	248	1.039	2
4	121	0.930	3
4	342	0.904	4
4	151	0.899	5
5	175	1.033	1
5	287	0.943	2
5	067	0.837	3
5	099	0.834	4
5	057	0.786	5

4.2. Location-Based Recommendation Model Building through K-Means Clustering

K-means clustering ascertains the “k” number of centroids within a dataset. After that, it assigns every data point with the closest cluster. These data points eventually end up being in the cluster with the nearest mean. In our approach, the purpose of applying K-Means clustering is to group similar users based on their respective locations. As the users get clustered, the top most searched or interacted item among that user group starts to get recommended to each user. Figure 4 shows auto-generated locations for users from different places in Islamabad, along with the corresponding cluster IDs. As one hovers above any cluster, it shows the most searched item in that cluster, which gets recommended to users of that cluster. For example, in one of the clusters, the count of searched property ID 689 is the highest. Since it is equal to the highest count for that cluster, all users falling within that cluster will get property ID 689 as the recommended property for view.

**Figure 4.** Location Based Clusters over Islamabad City.

4.3. Recommender System Validation

To validate the recommendations, one can simulate the user behaviour and fill in the possible or missing ratings or, in our case, the interactions a user might have with any prospective property items. The simulated values can then be further evaluated with error metrics such as the mean squared error to determine the deviation of predicted over observed values. The overall error of these values can provide us with an overview of the accuracy of our model.

Table 6 shows the generated matrix for interactions a user would likely have with property items and the MSE calculation for the overall matrix. Other methods for model validation can be performed through recall and precision. Both are very useful, as they show how accurate the recommendations are. However, the issue with recall and precision is that after applying these metrics, the recommended items are not sorted by their weighted value.

Table 6. Items matrix for prospective interactions.

	I1	I2	I3	I(N-2)	I(N-1)	I(N)
U1	3.656	3.504	3.488	...	3.545	3.534
U2	3.65	3.507	3.503	...	3.531	3.568
U3	3.601	3.450	3.452	...	3.483	3.504
U4	3.686	3.518	3.515	...	3.535	3.518
U5	3.708	3.549	3.548	...	3.582	3.595

Iteration: 100. Total Mean Squared Error = 337.6037.

MAP@k (Mean Average Precision at k) is an evaluation metric that considers the order of the recommended items as well. In our case, we have recommended 5 items to the set of 5 users, so in our case $k = 5$. We set an experimental environment for our group of five portal users; they were provided with a list of recommended items in the order generated by our recommender engine. The users interacted with certain items and provided verbal feedback on whether the generated recommendations were of interest. For example, the statistical accuracy of the recommendation engine is as follows for a given User 1.

- [1, 1, 1, 0, 0] where 1 stands for a correct recommendation such that the user interacted with it and 0 stands for a recommendation with which the user did not interact.
- [1/1, 2/2, 3/3, 2/3, 1/3] is the precision at k.
- $(1/5) [1/1 + 2/2 + 3/3 + 2/3 + 1/3] = 0.7999$ is the average precision at k.

The precision is higher for the first three items which were interacted with, but for the last two items with which the user did not interact, the precision falls. Therefore, for user 1, the average precision is almost 80%. Whereas for all sets of users, this will be the mean average precision and can be calculated by taking the average precisions' mean.

4.4. House Price Prediction Model

As previously mentioned, we have cross-compared and validated two property price prediction models (1) Multiple Linear Regression (2) Keras Regression. MLR is based on traditional regression techniques, whereas Keras has its basis in neural networks. We tested both approaches in a runtime Python environment. Figure 5 provides a high-level description of the procedure adopted for the model. Both models performed well with the given data and parameters. Still, the model with lower error rates and better variance score or coefficient of determination was chosen as the final model for deployment.

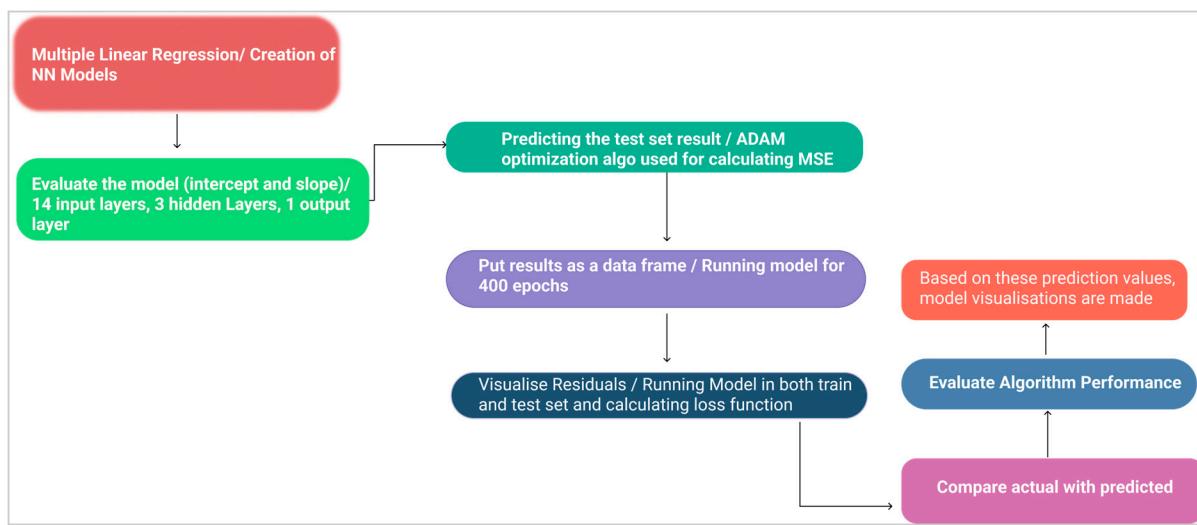


Figure 5. High-Level Description of Price Model.

4.4.1. Multiple Linear Regression

After running the first multiple linear regression model, Figure 6 illustrates the top-recommended properties based on location recommendations in different localities of Islamabad city, while Figure 7 represents the price prediction visualisation. The actual price is the one that was already present in the test data, whereas the predicted price was obtained after running the model on the train data. Table 7 represents the MAE, MSE and RMSE errors of these predictions. We can also see the variance score to be 0.70397 approximately.

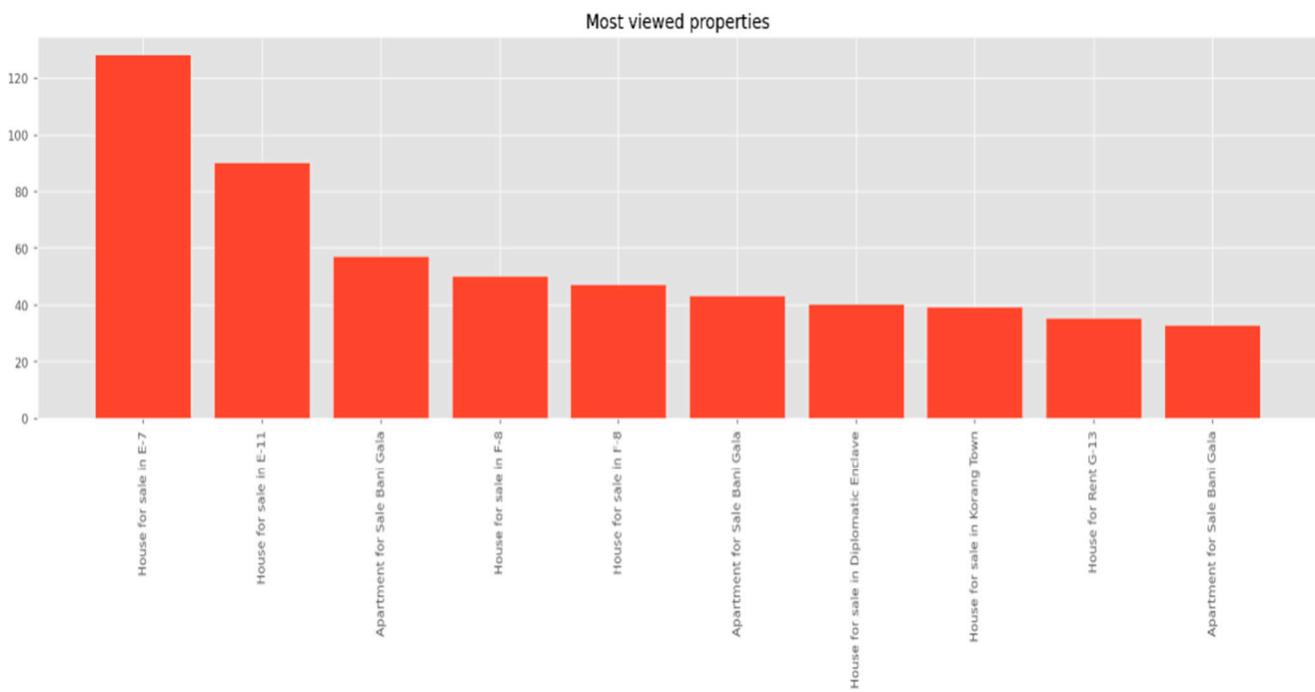


Figure 6. Top Recommended Properties Based on Location Recommendation in different localities of Islamabad City.

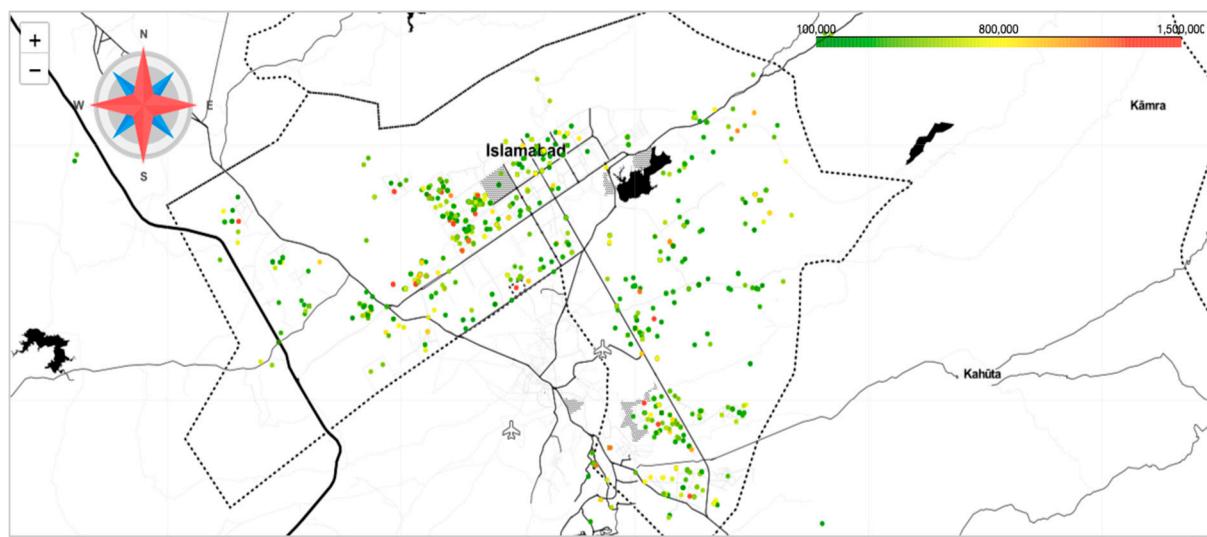


Figure 7. Actual vs. Predicted Price Visualisation of MLR.

Table 7. Complete Model Results: Multiple Linear Regression (Units in US dollars).

Evaluation Metric	Score
MAE	126,028.201
MSE	40,658,017,783
RMSE	201,638
VarScore	0.7039

4.4.2. Keras Regression

The variance score of Keras regression is approximately 0.8028. This is a better performance than the multiple linear regression approach. Furthermore, numerical errors for RMSE in the case of Keras have also been reduced, as shown in Tables 8 and 9. Therefore, our predictions in this case scenario are closer to actual pricing.

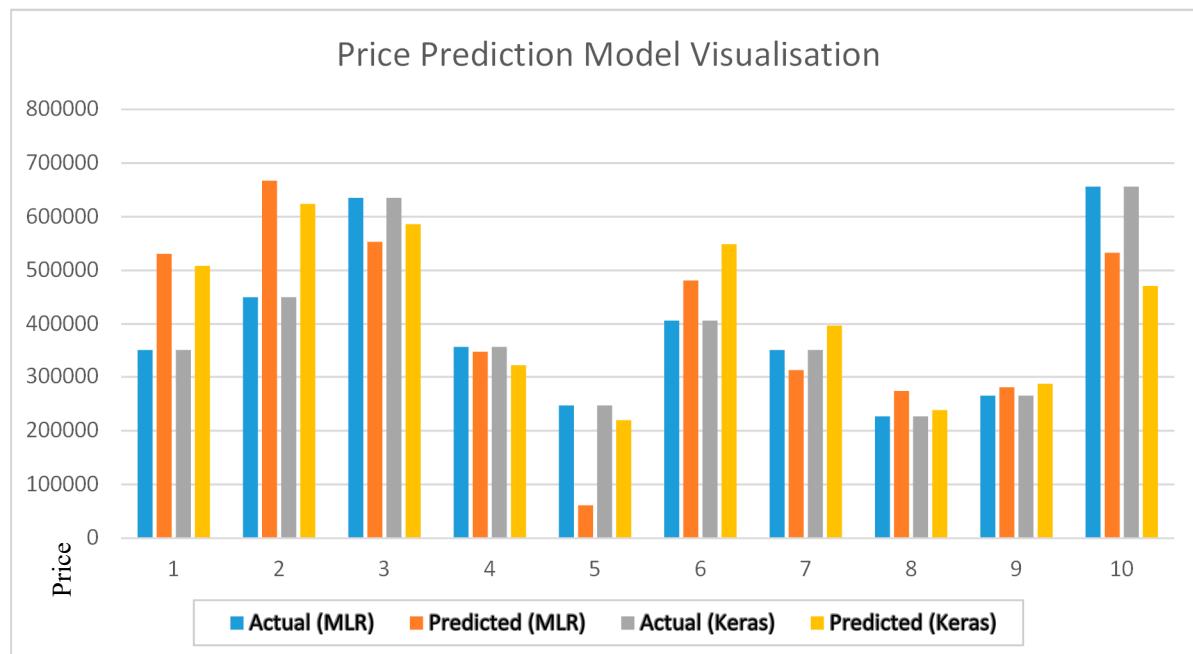
Table 8. Complete Model Results: Keras (Units in US dollars).

Evaluation Metric	Score
MAE	101,542
MSE	27,102,661,020
RMSE	164,628
VarScore	0.8028

Table 9. MLR and Keras Price Predictions (Units in US dollars).

Actual (MLR)	Predicted (MLR)	Actual (Keras)	Predicted (Keras)
349,950.0000	530,708.04458	349,950.0000	508,257.65625
450,000.0000	667,170.68394	450,000.0000	623,626.87500
635,000.0000	553,264.86718	635,000.0000	586,021.62500
355,500.0000	346,623.22842	355,500.0000	321,635.18750
246,950.00000	61,187.19574	246,950.00000	219,407.46875
406,550.00000	481,129.98291	406,550.00000	548,740.31250
350,000.00000	312,696.35790	350,000.00000	397,068.87500
226,500.00000	273,842.64629	226,500.00000	238,060.96875
265,000.00000	280,530.76516	265,000.00000	287,272.28125
656,000.00000	532,925.01517	656,000.00000	471,043.90625

Figures 8 and 9 represent different ways prices have been changing in the city of Islamabad. In Figure 8, mean sector (neighbourhood area) prices are highlighted. This visualisation provides a clear overview of what areas could show drastic price changes and what areas will remain stagnant. This data has been analysed for the past 2 years, and the predictions show how prices will change or remain the same in the coming years. The blue area in the figure depicts how prices have increased significantly in the coming years in those neighbourhoods. In contrast, red areas indicate stagnancy in prices, providing a user with a clearer picture to assist in decision making.

**Figure 8.** Price Prediction Visualisation over Islamabad City.

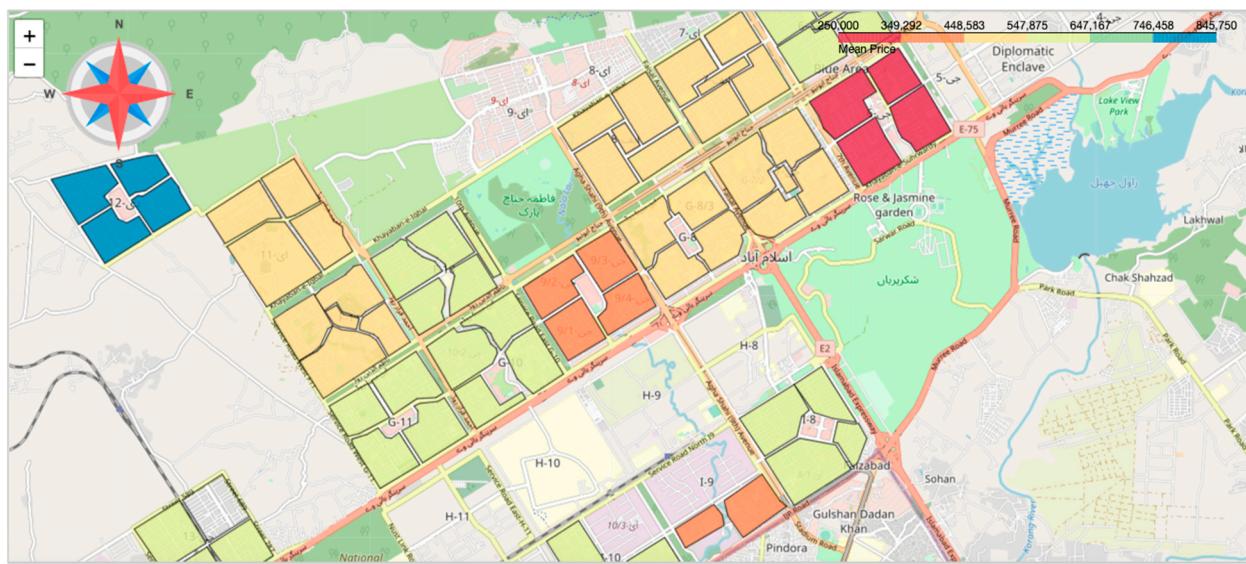


Figure 9. A visualisation of sector wise Predicted Price Visualisation in Islamabad City.

5. Conclusions

Three different recommendation algorithms for the real estate portal “Estatech Maps” were developed along with two different models for house price prediction. First, we set our goals to analyse and implement content-based filtering for suggesting real estate items. The collaborative filtering approach was used for reducing the computational cost by suggesting similar items to a similar group of users. Then, we applied the location-based approach for predicting the areas of interest to the user based on the user’s geographical location. All this was achieved with a minimum precision of 79%. Prediction models were created, and results were visualised by price increase, decrease or stagnancy in multiple sectors of Islamabad city to better assist people planning future land asset purchases. Our model was able to precisely predict the changes in house prices trends with a minimum accuracy of 80%, which was through our neural network-based prediction model. This work can be effectively utilised in any real-estate sale and purchase domain and will improve the overall user experience of real estate portals. This proves the viability of our map-based system in providing data and recommendations to users based on the popularity of an item, user similarity and geographical location.

While nowadays recommendation and predictive analysis are becoming a common trend in even the smallest of businesses, in Pakistan, the real estate industry is lacking when it comes to implementing these techniques not only in terms of a map-based interface but also in terms of presenting these items to the user in an effective way. Therefore, our approach for displaying an item of interest to the user on a map-based interface would be one of the pioneers in real estate portals in Pakistan.

We have used sequential NN models for our recommendation and prediction in this research. One area of improvement and basis for future work could be exploring and implementing these as parallel models to improve response time and efficiency. Another approach could be combining multiple techniques to create a hybrid model. The same approach was used in the study where the Cobb-Douglas and linear regression models were combined to form a mathematical model [24]. GIS was an additional tool to organise the regional data of the area under study. In turn, this can cover a broader spectrum of users’ behaviours and avoid high computational costs at the server end.

Author Contributions: Conceptualization, Maryam Mubarak, Ali Tahir and Fizza Waqar; methodology, Maryam Mubarak and Ali Tahir; software, Maryam Mubarak, Ali Tahir and Fizza Waqar; validation, Ali Tahir, Ibraheem Haneef, Gavin McArdle and Michela Bertolotto; formal analysis, Maryam Mubarak, Ali Tahir and Muhammad Tariq Saeed; investigation, Gavin McArdle, Maryam

Mubarak, Ibraheem Haneef, Ali Tahir and Muhammad Tariq Saeed; resources, Ali Tahir and Muhammad Tariq Saeed; data curation, Maryam Mubarak, Ali Tahir and Fizza Waqar; writing—original draft preparation, Maryam Mubarak, Ali Tahir and Fizza Waqar; writing—review and editing, Gavin McArdle, Michela Bertolotto and Muhammad Tariq Saeed; visualization, Maryam Mubarak, Fizza Waqar and Ali Tahir; supervision, Ali Tahir and Muhammad Tariq Saeed; project administration, Ali Tahir, Ibraheem Haneef and Muhammad Tariq Saeed; funding acquisition, Ali Tahir and Ibraheem Haneef. All authors have read and agreed to the published version of the manuscript.

Funding: This research received the funding from Higher Education Commission (HEC), Pakistan, under grant no. TDF03-249.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by the Higher Education Commission (HEC), Pakistan, under grant no. TDF03-249. The authors gratefully acknowledge their support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Geetha, G.; Safa, M.; Fancy, C.; Saranya, D. A hybrid approach using collaborative filtering and content based filtering for recommender system. *J. Phys. Conf. Ser.* **2018**, *1000*, 012101. [[CrossRef](#)]
2. Peter, D. Electronic junk. *Commun. ACM* **1982**, *25*, 163.
3. Knoll, J.; Groß, R.; Schwanke, A.; Rinn, B.; Schreyer, M. Applying Recommender Approaches to the Real Estate E-Commerce Market. In *International Conference on Innovations for Community Services*; Springer: Cham, Switzerland, 2018; pp. 111–126.
4. Bai, Y.; Jia, S.; Wang, S.; Tan, B. Customer Loyalty Improves the Effectiveness of Recommender Systems Based on Complex Network. *Information* **2020**, *11*, 171. [[CrossRef](#)]
5. Fernández-García, A.J.; Iribarne, L.; Corral, A.; Criado, J.; Wang, J.Z. A recommender system for component-based applications using machine learning techniques. *Knowl. Based Syst.* **2019**, *164*, 68–84. [[CrossRef](#)]
6. Rabiei-Dastjerdi, H.; McArdle, G.; Matthews, S.A.; Keenan, P. Gap analysis in decision support systems for real-estate in the era of the digital earth. *Int. J. Digit. Earth* **2021**, *14*, 121–138. [[CrossRef](#)]
7. Mac Aoidh, E.; Bertolotto, M.; Wilson, D.C. Understanding geospatial interests by visualizing map interaction behavior. *Inf. Vis.* **2008**. [[CrossRef](#)]
8. Yu, Y.; Wang, C.; Zhang, L.; Gao, R.; Wang, H. Geographical Proximity Boosted Recommendation Algorithms for Real Estate. In *International Conference on Web Information Systems Engineering*; Springer: Cham, Switzerland, 2018; pp. 61–66.
9. Breese, J.S.; Heckerman, D.; Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, USA, 24–26 July 1998; pp. 43–52.
10. Shani, G.; Heckerman, D.; Brafman, R.I. An MDP-based recommender system. *J. Mach. Learn. Res.* **2005**, *6*, 1265–1295.
11. Ayyaz, S.; Qamar, U.; Nawaz, R. HCF-CRS: A Hybrid Content based Fuzzy Conformal Recommender System for providing recommendations with confidence. *PLoS ONE* **2018**, *13*, e0204849. [[CrossRef](#)] [[PubMed](#)]
12. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
13. Ballatore, A.; McArdle, G.; Kelly, C.; Bertolotto, M. Recomap: An interactive and adaptive map-based recommender. In Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, 22–26 March 2010; pp. 887–891.
14. Wilson, D.C.; Lipford, H.R.; Carroll, E.; Karr, P.; Najjar, N. Charting New Ground: Modeling User Behavior in Interactive GeoVisualization. In Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Irvine, CA, USA, 5–7 November 2008; pp. 1–4.
15. Tezuka, T.; Tanaka, K. Presentation of Dynamic Maps by Estimating User Intentions from Operation History. In *International Conference on Multimedia Modeling*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 156–165.
16. Rehman, F.; Masood, H.; Ul-Hasan, A.; Nawaz, R.; Shafait, F. An Intelligent Context Aware Recommender System for Real Estate. In *Mediterranean Conference on Pattern Recognition and Artificial Intelligence*; Springer: Cham, Switzerland, 2019; pp. 177–191.
17. Kong, J.S.; Teague, K.; Kessler, J. The Love-Hate Square Counting Method for Recommender Systems. *Proc. KDD Cup* **2011**, *18*, 249–261.
18. Greenaway-McGrevy, R.; Sorensen, K. A spatial model averaging approach to measuring house prices. *J. Spat. Econom.* **2021**, *2*, 1–32. [[CrossRef](#)]
19. Rawool, A.G.; Rogye, D.V.; Rane, S.G. House Price Prediction Using Machine Learning. *Int. J. Res. Appl. Sci. Eng. Technol.* **2021**, *9*, 686–692. [[CrossRef](#)]
20. Chaturvedi, S.; Ahlawat, L.; Patel, T.; Talha, M. Real Estate Price Prediction. *EasyChair* **2021**, 4926. Available online: <https://easychair.org/publications/preprint/HbD8> (accessed on 2 January 2022).

21. Abdullahi, A.; Usman, H.; Ibrahim, I. Determining house price for mass appraisal using multiple regression analysis modeling in Kaduna North, Nigeria. *ATBU J. Environ. Technol.* **2018**, *11*, 26–40.
22. Renigier-Bilozor, M.; Źróbek, S.; Walacik, M.; Borst, R.; Grover, R.; d'Amato, M. International acceptance of automated modern tools use must-have for sustainable real estate market development. *Land Use Policy* **2022**, *113*, 105876. [[CrossRef](#)]
23. Mubarak, M.; Khalid, K.; Waqar, F.; Tahir, A.; Haneef, I.; McArdle, G.; Bertolotto, M. Towards Real Estate Analytics using Map Personalisation. In Proceedings of the 6th International Conference on Geographical Information Systems Theory, Applications and Management, GISTAM 2020, Prague, Czech Republic, 7–9 May 2020; pp. 184–190.
24. Sisman, S.; Akar, A.U.; Yalpir, S. The novelty hybrid model development proposal for mass appraisal of real estates in sustainable land management. *Surv. Rev.* **2021**, *1*–20. [[CrossRef](#)]