TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Software Science

Dmitri Batõrjev 144119IAPM

POINT OF INTEREST ATTENDANCE PREDICTION WITH LIMITED DATA ACCESS

Master's thesis

Supervisor: Ago Luberg
MSc

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Dmitri Batõrjev 144119IAPM

TURISMIOBJEKTIDE KÜLASTATAVUSE ENNUSTAMINE KASUTADES PIIRATUD LIGIPÄÄSUGA ANDMEALLIKAT

Magistritöö

Juhendaja: Ago Luberg
MSc

Tallinn 2017

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Dmitri Batõrjev

28.12.2016

Abstract

The main purpose of this work is to develop an approach to predict attendance at different POIs with limited data access. To solve this kind of a problem we consider and apply different machine learning techniques. The whole process consists of data collection and preparation, cluster analysis of POIs and prediction model building.

The given approach can be used in bigger tourism recommender systems to provide users with better recommendations depending on user's preferences and desired period of time. Besides that, the given approach can be applied in other domains, for example, to predict attendance at special events and so on.

This thesis is written in English and is 45 pages long, including 10 chapters, 11 figures and 5 tables.

Annotatsioon

TURISMIOBJEKTIDE KÜLASTATAVUSE ENNUSTAMINE KASUTADES PIIRATUD LIGIPÄÄSUGA ANDMEALLIKAT

Antud töö eesmärk on arendada lähenemisviis, mis võimaldab ennustada sündmuskohtade külastatavust piiratud ligipääsuga andmetega. Selle eesmärgi saavutamiseks käsitleme ja kasutame erinevaid masinõppe meetodeid. Kogu protsess koosneb andmete korjamisest ja töötlemisest, sündmuskohtade klasterdamisest ning ennustusmudelite loomisest.

Selle töö tulemus on lähenemisviis, mida on võimalik kasutada turismi soovitussüsteemides selleks, et pakkuda kasutajatele paremat soovitust sõltuvalt kasutaja eelistusest. Antud lähenemisviisi on samuti võimalik kasutada teistes valdkondades, näiteks, et ennustada sündmuste külastatavust jne.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 45 leheküljel, 10 peatükki, 11 joonist, 5 tabelit.

List of abbreviations and terms

POI	*Point of interest*
API	*Application programming interface*
REST	*Representational state transfer*
JSON	*JavaScript Object Notation*
GDP	*Gross domestic product*
MATLAB	*Numerical computing environment*
RMSE	*Root Mean Square Error*
MSE	*Mean Square Error*
MAE	*Mean Absolute Error*

Table of contents

List of figures

List of tables

1 Introduction

Tourism is one of the most popular activities. There were 1184 million international tourist arrivals worldwide in 2015, which is 4.4% more than a year before, and this number keeps increasing [1]. For many countries and regions the tourism is a major source of income. In 2012 26.5% of total GDP contribution of Aruba was the tourism, in Macau this number was 46.7% [2]. With the growth of the tourism the number of points of interest (POI) also grows and it becomes harder for tourists to choose a destination and plan a trip. There is a great number of different companies and online services that offer recommendations for tourists: Foursquare, TripAdvisor, TripExpert, etc. These services help tourists to exchange information, share opinions about different places and get recommendations.

One of the recommendation systems is http://www.sightsmap.com. This system helps tourists to plan their trip. A tourist selects start and end points of the route, chooses desired time and a movement method (walk or drive) and the system provides a route based on the tourist's preferences. To improve the recommendation system and provide the most suitable trip to the users depending on the time of the trip we can use visiting history to predict popularity of the objects in the future. This will help us to choose between multiple objects in case a tourist does not have enough time to visit all the available places on the route.

The main purpose of this work is to develop an approach to predict attendance at different POIs using limited amount of data. To solve this kind of a problem we consider and apply different machine learning techniques. The result of this work can also be applied in other domains.

1.1 Goals

The goals of this work is to:

1. Collect and process the visiting information
2. Cluster places
3. Build predictive models
4. Evaluate the result

1.2 Methodology

In this work Foursquare is used as a source of information. After enough data is collected and processed, different machine learning techniques are used to analyze the data and create models that can be used for a prediction. The whole process consists of the following steps:

1. Data collection
2. Data preparation
3. Clustering
4. Prediction

Data collection and preparation is done by applications written in Java. All the data is stored in PostgreSQL database. Data analysis, visualization, clustering and prediction are done in MATLAB software.

1.3 Overview

To be able to predict attendance at some POI at specific time we need its history of visits. As mentioned above, we collect the data from Foursquare. This service provides only information about the total number of check-ins (visits) made at specific POI to date, while we are interested in information about check-in changes. In order to know check-in changes, the visiting information must be constantly requested and the difference between two moments must be calculated. Because of the rate limits discussed in the

section 2.1, the information cannot be requested very frequently. This is the reason we decided to collect the changes every hour.

The rate limits also prevent us from collecting enough data to build a predictive model. We assume that the attendance at different POIs can be similar, therefore it was decided to cluster the POIs and use the visiting history of all the places in the cluster to build a predictive model. The clustering can be made basing on common information of the POIs such as a category, location, etc. But we suppose that the attendance at same category objects within same location can vary, therefore we decided to cluster the POIs basing on visitors' behaviour at these places. Knowing when some object is visited and when it is not, allows us to compose a so-called attendance profile, which represents information about visits in specific hours.

After the POIs are clustered, all the visiting information of all places in the cluster is used to build a predictive model for each cluster. These models are then used to predict a number of visitors at some place.

For example, there are two places that are visited only on Fridays from 22 to 23 o'clock. Last Friday these places were visited by 10 and 4 users respectively during this time. Despite the fact the number of visitors is different, these places have similar attendance - they are only visited on Fridays from 22 to 23 o'clock, so the places can share a cluster. After the data is normalized and places are clustered, it is possible to build a predictive model for each cluster using visiting history of both places. These models can then be used to predict how many visitors will be there on next Friday at the places. The whole process is displayed in Figure 1.

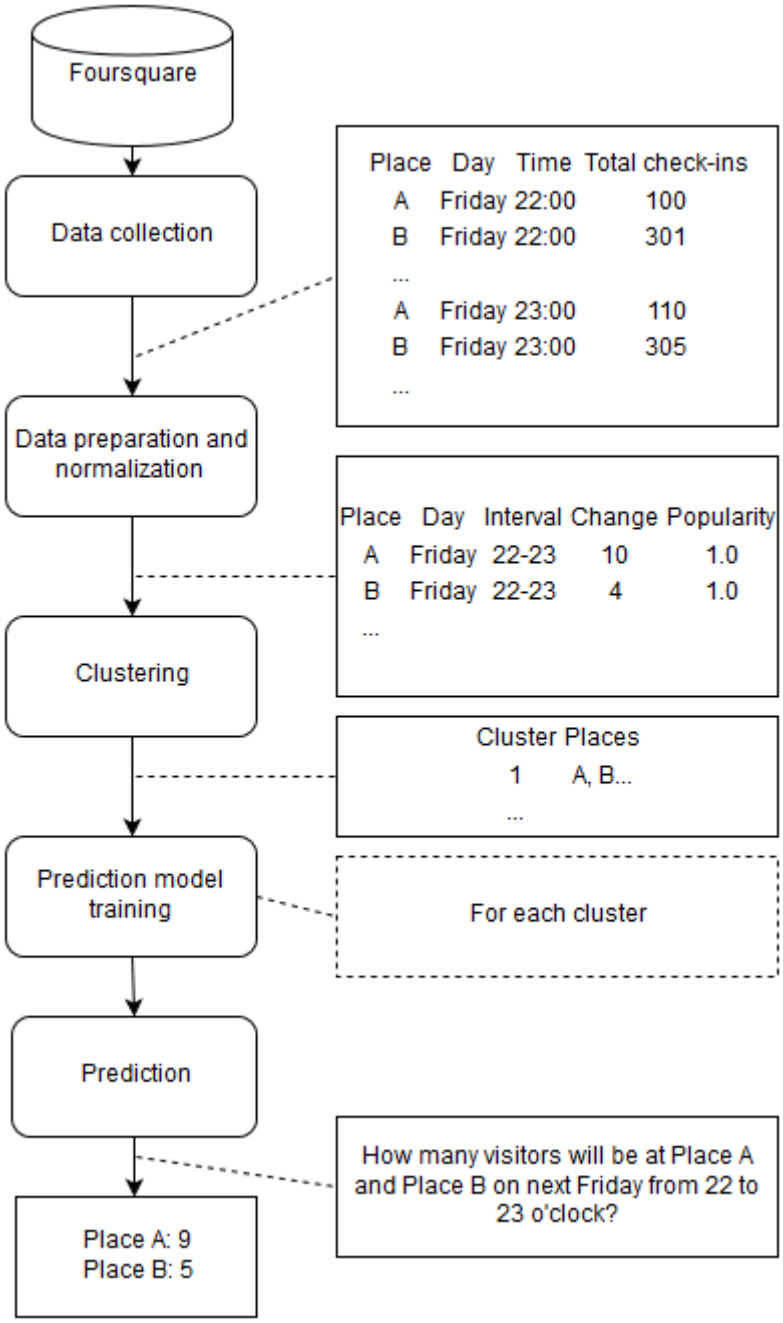

Figure 1. Attendance prediction process.

2 Data source

Foursquare is a service that provides information about different places around the world. Foursquare allows users to search places, rate them and leave or view tips from others. The service was launched in 2009 and in the beginning it was more like a social network where users could mainly share their location information with friends. Five years later, in 2014, Foursquare Labs, Inc., the owners of the Foursquare application, introduced a new application called Swarm. The location sharing functionality in Foursquare was moved to Swarm and Foursquare became a place information and recommendation providing service [3].

Foursquare and Swarm are highly popular applications with over 50 million active users and over 10 billion check-ins worldwide. The applications also provide information about a great number of places. By now this number is 93 million places worldwide [3].

2.1 Available data and restrictions

Like many other social networks, Foursquare allows developers to access their and their users' public data through an API. It is possible to retrieve information about places, users, check-ins (although the location sharing functionality was moved to the Swarm application, this kind of data is available only through the Foursquare API) and so on.

The information we are interested in is venue (place) information and their statistics. The venue's statistics contain the information about how many people visited the specific venue, a total number of check-ins and a number of users that are there at the moment.

One of the problems is that check-ins chronological information is only available to a place's owner. Others can see only a total number of check-ins at some place at the moment. As we are interested in information about check-in changes, we need to constantly request places' information and compute the difference.

Another problem is a rate limit and a limited number of items in the response. It is allowed to do maximum 5,000 request per hour to the API and a response contains only up to 50 places. The rate limit also means that maximum 120,000 areas can be requested per day. Because of this Foursquare can be used only to track limited number of areas. Currently the following cities (and their neighbourhoods) are being constantly requested: Madrid,

Rome (also Vatican), Berlin, Barcelona, Tallinn, Paris, Hamburg, Budapest, London, Amsterdam, Prague, Vienna, Venice, Dublin, Lisbon, Athens and Florence. These cities were selected, because they are the most populated cities in Europe [4]. The areas mentioned previously are requested at the beginning of each hour.

To reduce the time between requests, responses are processed in parallel in another thread. It takes about 16-18 minutes to make all the 5,000 requests. Ideally requests must be done within some small time window (e.g. 14:55-15:05) to acquire the most accurate hourly visiting information. Requesting the areas within a small window at the beginning of each hour allows us to easily calculate the number of check-ins that were made during this hour.

2.2 API

The Foursquare API is based on a REST architectural style.

To retrieve the places' and check-ins' information the following endpoint is used:

https://developer.foursquare.com/docs/venues/search

Places can be found basing on coordinates, city name and other query parameters. In our case, we use south-west and north-east coordinates of the cities to retrieve places.

The following request parameters are being used:

Table 1. Foursquare API request parameters.

Parameter name	**Description**	**Value (we use)**
intent	Intent of the search that tells Foursquare how to perform the search. For our purposes we use „browse“ value, meaning that Foursquare will return venues located in given area.	browse
sw	Comma-separated south-west latitude and longitude	-90...90,-180...180
ne	Comma-separated north-east latitude and longitude	-90...90,-180...180

Parameter name	Description	Value (we use)
limit	Number of results to be returned. Maximum allowed value is 50.	50
client_id	Client's identificator generated by Foursquare	
client_secret	Client's secret key generated by Foursquare	
v	Version of the client (application)	20160301

If the request is valid and no errors occurred, Foursquare returns a list of places in a JSON format. The response structure is described below.

Table 2. Foursquare API response parameters.

Parameter name		Description
id		Foursquare unique string identifier of the Place
name		Name of the place
location	country	Country the place is located in
	city	City the place is located in
	address	Address of the place
	lat	Latitude of the place
	lng	Longitude of the place
categories		List of categories. Each category has an identifier and a name
stats	checkinsCount	Total number of check-ins
	usersCount	Total number of unique check-ins

In case the rate limit is reached or Foursquare servers are unavailable, the program stops and waits for 20 seconds and then continues to request areas' information.

2.3 Area division

As mentioned above, places' information is requested basing on south-west and north-east coordinates of the area. The problem is that Foursquare returns information about up to 50 places, which means that there can be more places in the area. Foursquare also returns an error if the requested area is bigger than allowed (areas up to approximately 10,000 square kilometres are currently supported). A solution to these problems is to divide the areas in case there is 50 places in the response or the response contains the error listed above. The division is made basing on coordinates (degrees).

```
areas = [initial area]
while true
	area = areas.getNextArea()
	places = getPlacesFromFoursquare(area)
	if places.size = 50 then
		if abs(area.eastCoordinate – area.westCoordinate) >
abs(area.southCoordinate – area.northCoordinate) then
		    [area1, area2] = divideVertically(area)
		otherwise
		    [area1, area2] = divideHorizontally(area)
		end
		areas.add(area1, area2)
	end
end
```

Figure 2. Area division algorithm.

3 Data preparation

After the data is collected, it needs to be prepared to be used in MATLAB software. We retrieve visiting information at some moments, so we have to calculate the difference between two dates. Also we have to apply time zone information and extract the hour of the day and the day of the week, which are the features we are using to cluster the places and build predictive models.

For example, if we requested some object's information on 09.09.2016 at 13 and at 14 o'clock, and the total number of check-ins were 4 and 10 respectively, the application must output the following data point:

5 (Friday), 13 (from 13 to 14), 6 (visits)

3.1 The data

In this work it was decided to use information only of these places that have average 50 or more check-ins per week. This filter will remove abandoned and unpopular places, which most likely are not related to tourism.

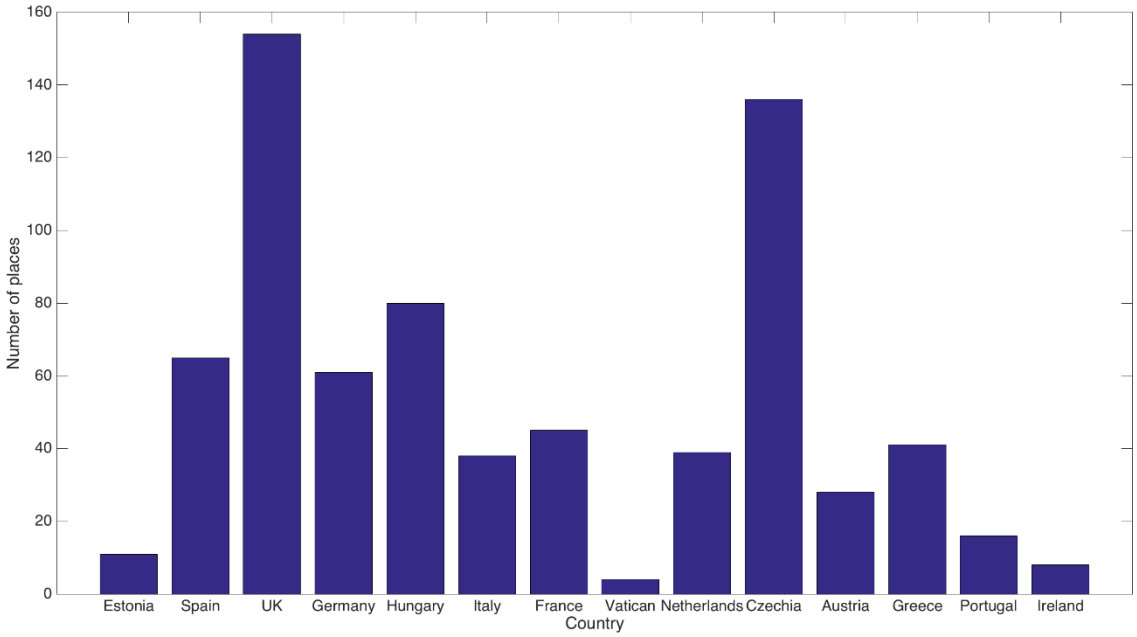

Figure 3. Distribution of the places among countries in the data set.

The final data set contains 185,413 rows of check-in changes of 726 different places from 14 countries. The distribution of the places among the countries is shown in Figure 3.

67.7% (125,445) of the data are zero check-ins (meaning there were not any visitors during these hours), 15.2% (28,151) - single check-ins.

The number of check-in distribution in the data set is shown in Figure 4.

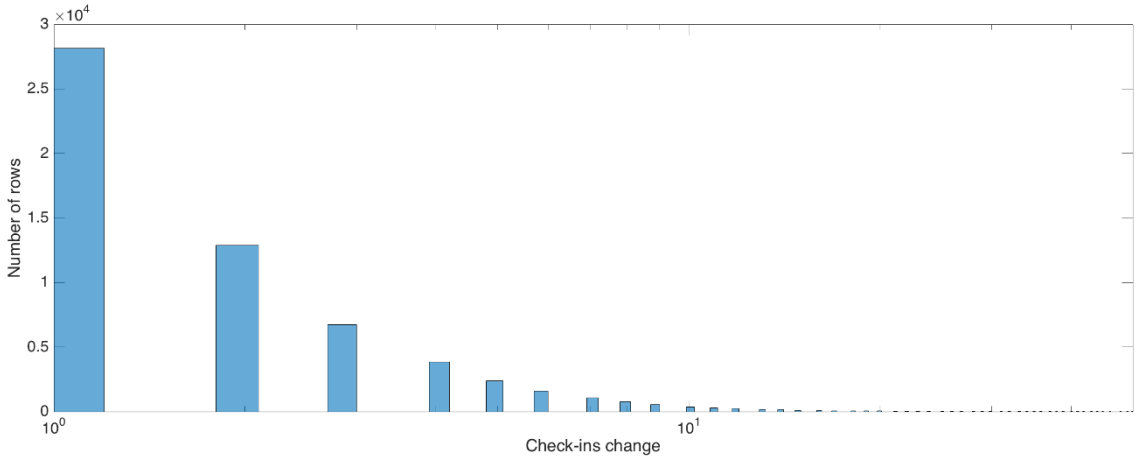

Figure 4. Histogram of check-ins changes in the data set.

The next histogram shows the distribution of the total number of check-ins.

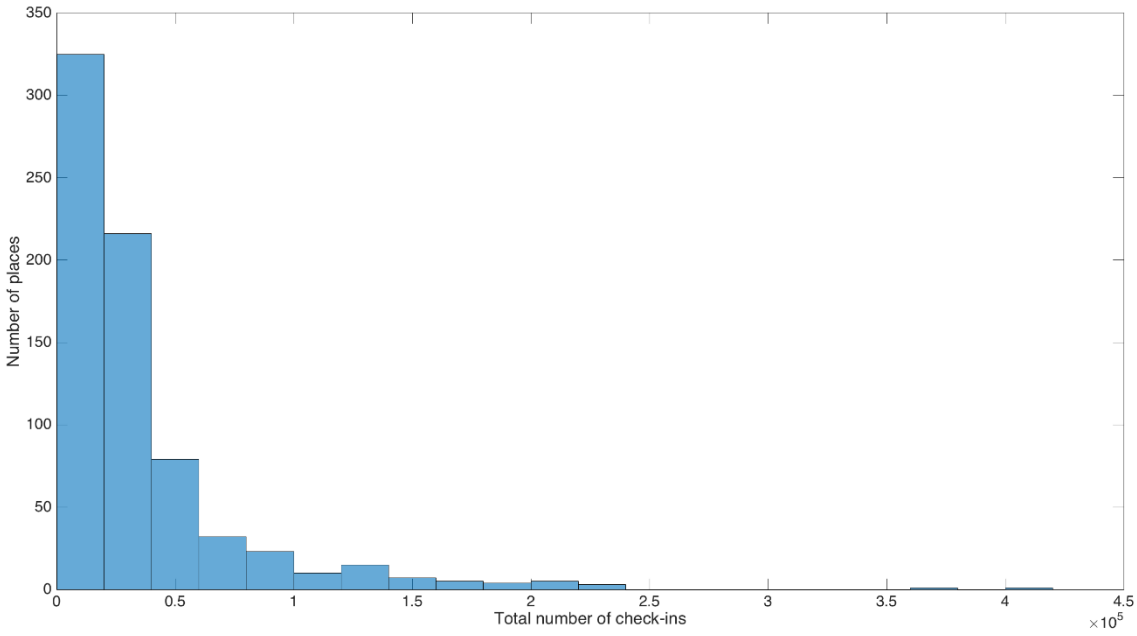

Figure 5. Histogram of total number of check-ins.

4 Clustering

There are different ways to solve the problem of predicting popularity of different POIs. For example, we can build a separate model for each object or we can cluster the objects and then build a model for each cluster. Clustering can be made based on object's common information like a category, location and so on, or it can be done based on visitors' behaviour at these places. Knowing when some object is visited and when it is not, allows us to compose a so-called attendance profile and cluster the objects into groups using this profile. Building a separate model for each object will give us more accurate predicting models, but it will take much more time for models to converge, because there is a great amount of objects we want to follow and the Foursquare service has some limitations that we mentioned earlier. So in this work it was decided to cluster the objects and then build a predicting model for each cluster separately. Clustering will be made using an attendance profile, because we assume that popularity of same category objects within the same location can vary. For example, there are regular and night gyms, some tourist objects are popular in the late evening, others are visited only in the mornings, etc.

4.1 Popularity and data normalization

Different places have a different number of visitors. Some places are visited by hundreds of people weekly, others are visited by a much smaller number of visitors. If we want to cluster these places and predict their popularity later, we need to define some coefficient that will allow us to unite the places. For example, the objects O1 and O2 are only visited on Tuesdays from 7:00 to 8:00 and during this time have 10 and 80 visitors (check-ins) respectively. Despite the fact that the number of check-ins is different, these places have similar attendance - they are only visited on Tuesdays from 7:00 to 8:00, so they can share one model to predict popularity in future.

Here we define the popularity coefficient as follows:

$$P_t = \frac{C_t}{mean(C_{nz})}$$

Where:

t - Time, 1-hour interval (Tuesday from 8 to 9 o'clock, etc.)

P_t - Object's popularity during the hour

C_t - Number of check-ins were made during the hour

C_{nz} - Object's all non-zero check-ins

Here we use non-zero check-ins because the places have different opening hours and the coefficient of the places that have shorter opening hours should not suffer.

This coefficient shows how many times the attendance at the specific place is larger than usual. Zero means it is empty at given time, a value between zero and one shows that the attendance is below the average, one means usual activity and a value greater than one shows that the place is more popular than usual.

4.2 K-means

There are many different clustering techniques we could use. In the given work we decided to use the same approach as in the related work [5] and to cluster the objects using the k-means algorithm with the squared Euclidean distance. The k-means is an iterative clustering algorithm. Given k randomly selected initial centroids, the algorithm calculates distances between points and the centroids and assigns the points to the nearest centroid. After all the points are assigned to the centroids, the algorithm updates the centroids with a mean value of the points assigned to the centroid. The assignment and centroid update are repeated until no changes can be made in clusters.

4.3 Squared Euclidean distance

The difference between the Squared Euclidean distance and the regular Euclidean distance is an absence of square root in the equation. This makes the clustering faster without affecting the result [6].

$$d(A, B) = (A_1 - B_1)^2 + (A_2 - B_2)^2 + \ldots + (A_n - B_n)^2$$

Where:

d(A, B) - Distance function

A, B - Position of a point, n-length vector

4.4 Attendance profile

As we already mentioned above, we will cluster the place using their attendance profiles instead of using different attributes of the places like category, coordinates, etc. The attendance profile of the object represents average hourly popularity on each day. So a profile consists of 7 * 24 = 168 features. Later we can also add other features like popularity on holidays, season popularity and so on. The more features we will use, the more accurate the prediction will be, but this also means that we will need to track check-in changes for longer time.

4.5 Feature selection

168 is quite a big amount of features and adding new features will make it problematic to cluster a large number of places as it will decrease the speed of the clustering. Also, using all possible hours and days of the week makes it really problematic to classify new places, because it means that we have to collect object's check-ins changes for all these hours and days before we can attach this object to some existing cluster. To reduce the dimensionality we decided to use only equidistant features with high variance [7]. We need to select equidistant features, because different places have different opening hours and using only features with the highest variance will affect the predictive models. The variance of the features is displayed in Figure 6.

The following hours were chosen: 0:00-1:00, 5:00-6:00, 9:00-10:00, 13:00-14:00, 17:00-18:00, 19:00-20:00.

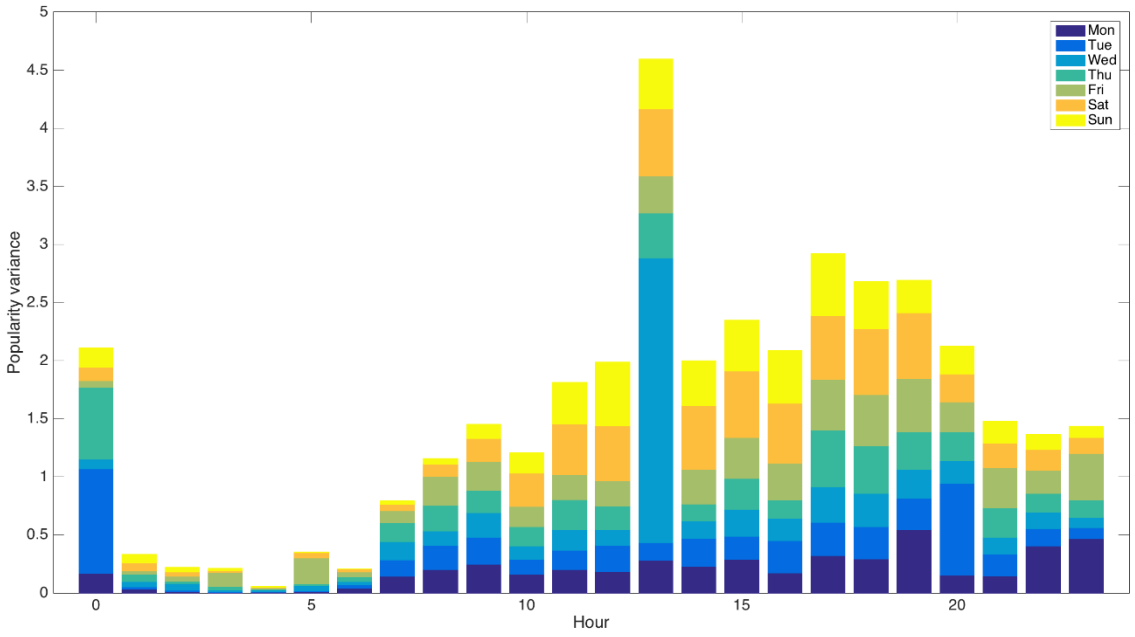

Figure 6. Variance of the features.

4.6 Number of clusters

It is a common problem to choose the number of clusters (especially when there are a lot of dimensions) and there are a lot of different techniques we can use: elbow method, silhouette, etc. In the given work to determine an appropriate number of clusters we decided to use the elbow method, which is a visual method [8]. The main idea is to cluster the data several times starting with k=2 increasing the total number of clusters each time. At the beginning the cost of the clustering drops dramatically, but at some point the cost of the clustering starts to decrease much slower and reaches a so-called plateau. On the plot this point looks like an elbow, which shows an appropriate number of clusters [8].

After clustering the objects 149 times with different value of k (from 2 to 150) and calculating the sum of the distances of each object to its cluster's centroid (clustering cost function), we got the result shown in Figure 7.

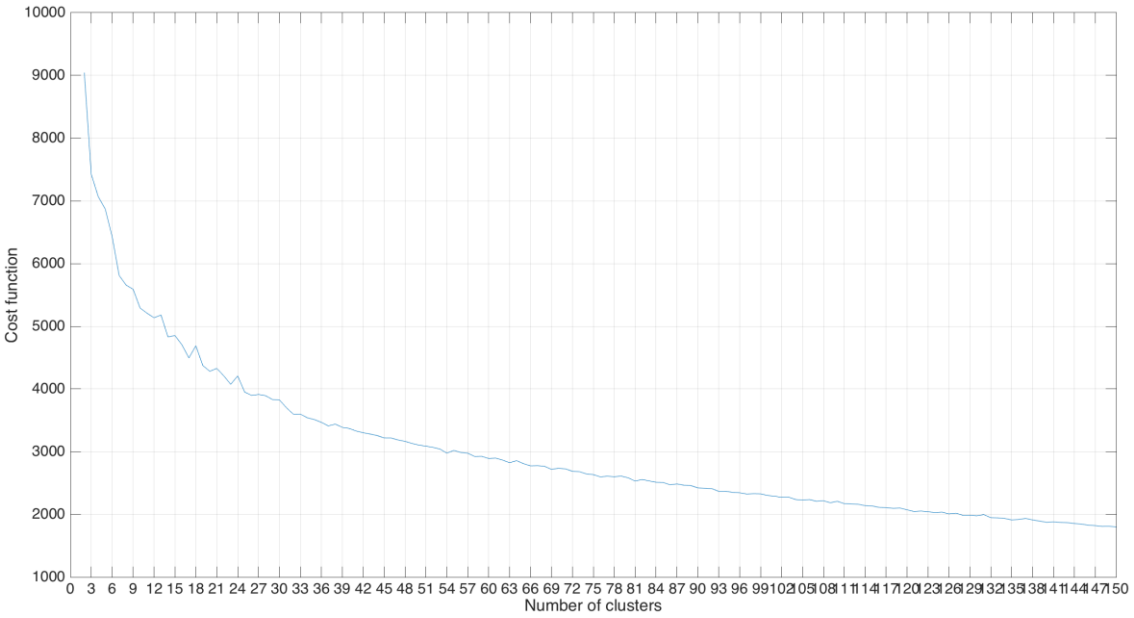

Figure 7. Cost function value of the clusters.

The figure shows that the appropriate number of clusters is between 25 and 30. This is the range where the cost of the clustering reaches the plateau. We decided to use 25 clusters.

4.7 Clusters

After clustering objects into 25 clusters, we got the following result:

Table 3. Clusters. Top categories and counties.

Cluster	**Number of places**	**Top category (number of occurrences)**	**Top county (number of occurrences)**
1	1	Hockey Arena (1)	Czechia (1)
2	204	Neighborhood (21)	United Kingdom (42)
3	2	Soccer Stadium (1)	Spain (1)
4	80	Metro Station (10)	Czechia (19)
5	1	Train Station (1)	Austria (1)
6	56	Neighborhood (8)	United Kingdom (20)
7	71	Neighborhood (7)	Hungary (15)
8	1	Plaza (1)	Spain (1)
9	1	Plaza (1)	Czechia (1)
10	1	Neighborhood (1)	United Kingdom (1)

Cluster	Number of places	Top category (number of occurrences)	Top county (number of occurrences)
11	1	Neighborhood (1)	United Kingdom (1)
12	31	Neighborhood (7)	United Kingdom (14)
13	1	Light Rail Station (1)	Germany (1)
14	18	Plaza (3)	Czechia (9)
15	39	Metro Station (13)	United Kingdom (16)
16	32	Multiplex (5)	Czechia (15)
17	1	French Restaurant (1)	United Kingdom (1)
18	34	Shopping Mall (8)	United Kingdom (9)
19	32	Shopping Mall (5)	Spain (7)
20	1	Shopping Mall (1)	Czechia (1)
21	26	Neighborhood (3)	Czechia (7)
22	11	Hotel (4)	United Kingdom (6)
23	3	Clothing Store (1)	United Kingdom (2)
24	25	Train Station (15)	United Kingdom (8)
25	53	Plaza (7)	Hungary (11)

The 11 of the 25 clusters have a small amount (from one to three) of objects. This is because they had unusual attendance at some day. For example, on October 22 from 19:00 to 20:00 (local time) at O2 arena, the only place from cluster 1, 68 check-ins were made while the average hourly check-ins change at this place is 5.3. The restaurant Aubaine (cluster 17) also had an unusual attendance on September 26 from 00:00 to 01:00, when 129 check-ins were made (average hourly check-ins change is 4.7). We assume that some special events took place there at these days. Because of this, these places could not share a cluster with any other places. The given problem could be solved by collecting the information about different events happening at the places and using it in attendance profiles. This also means that the check-ins changes have to be tracked for a longer period of time to fully compose an attendance profile, which also covers the attendance on special days.

The full table describing all the countries and categories within the clusters is in Appendix 1 of the document.

Table 4. Clusters. Most popular day of the week and hour of the day.

Cluster	**Most popular day**	**Most popular hours (among those that were used in the clustering)**	**Centroid value (mean popularity)**
1	Saturday	19:00-20:00	12.75
2	Saturday	19:00-20:00	0.14
3	Thursday	0:00-1:00	14.4
4	Friday	17:00-18:00	0.97
5	Tuesday	0:00-1:00	24.36
6	Saturday	13:00-14:00	1.21
7	Tuesday	19:00-20:00	0.92
8	Wednesday	13:00-14:00	40.81
9	Thursday	17:00-18:00	11.73
10	Sunday	17:00-18:00	9.21
11	Monday	19:00-20:00	16.28
12	Saturday	17:00-18:00	1.82
13	Friday	5:00-6:00	11.82
14	Friday	19:00-20:00	2.83
15	Friday	9:00-10:00	1.32
16	Thursday	19:00-20:00	1.92
17	Monday	0:00-1:00	9.18
18	Wednesday	17:00-18:00	1.44
19	Saturday	17:00-18:00	2.42
20	Thursday	13:00-14:00	11.76
21	Sunday	17:00-18:00	2.14
22	Sunday	0:00-1:00	2.38
23	Saturday	13:00-14:00	5.39
24	Thursday	9:00-10:00	1.67
25	Saturday	13:00-14:00	1.63

All the other clusters have different peak times, when the objects are visited mostly. Majority of the clusters are popular at the end of the week. The most popular hours (among those that were used in the clustering) are in the evening from 17 to 18 o'clock.

To evaluate the clusters, we plotted the attendance at two random places within four different clusters.

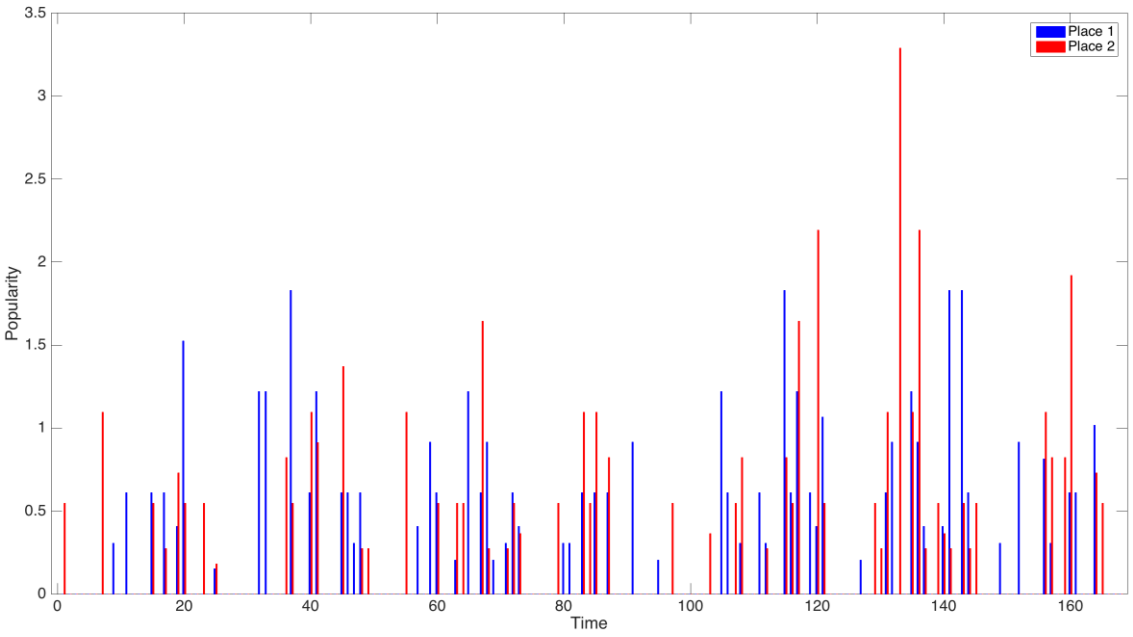

Figure 8. Popularity of two random places from cluster 2.

The figures show that not all the places within the cluster have completely identical attendance. This is because we are not using all the available 168 features to cluster the places.

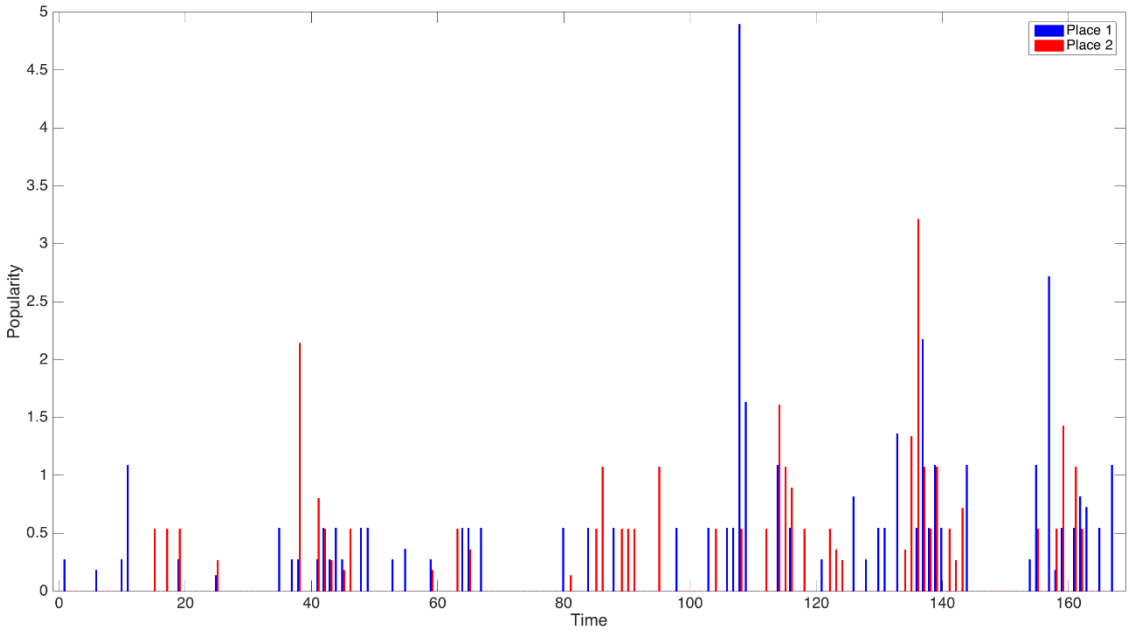

Figure 9. Popularity of two random places from cluster 4.

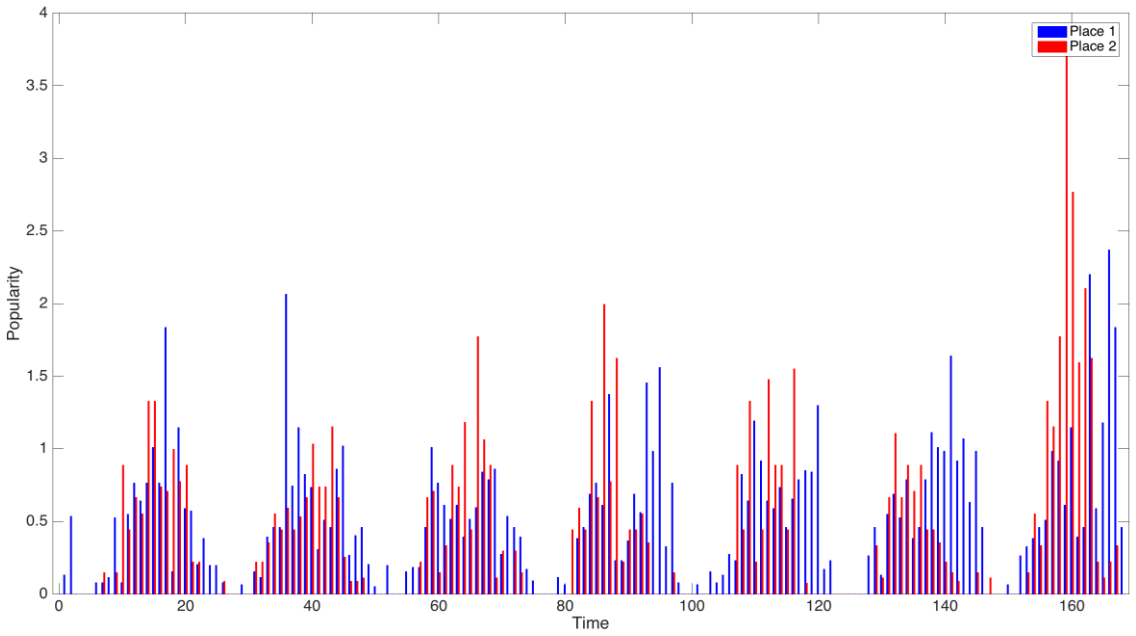

Figure 10. Popularity of two random places from cluster 6.

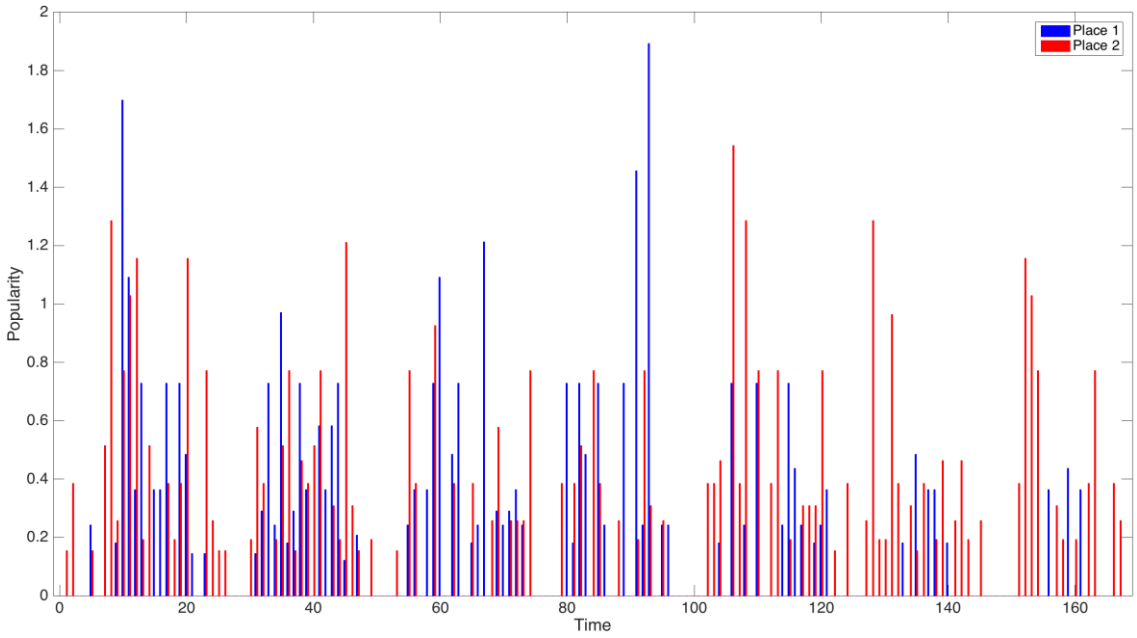

Figure 11. Popularity of two random places from cluster 15.

4.8 Classification

When the number of places is relatively small, clustering is done quickly and it is possible to run the clustering every time new places are added. But at some point the clustering will not be so efficient and will take much more time. One of the solutions is to classify the new places. After the attendance profile of the new place is computed, it can be used for classification using the distance weighted k-NN algorithm. This is an advanced k-NN

algorithm, where the distance between the points plays an important role. The nearest points have the biggest vote for the class (cluster) [9].

The steps of the given algorithm are following:

1. Calculate distances between a new point and all other points
2. Select k number of nearest points (neighbours)
3. Assign a class to the point basing on contribution of each neighbour

$$F(x_q) = \arg\max_{v \in V} \sum_{i=1}^{k} w_i \, \delta(v, f(x_i))$$

Where:

x_q – Query point
V – Set of classes
k – Total number of points
$\delta(a, b)$ = 1 if $a = b$, otherwise $\delta(a, b) = 0$

And where the weight is:

$$w_i = \frac{1}{d(x_q, x_i)^2}$$

If the distance between a new and some other points is zero (exact match), the class of the identical point is assigned to the new one.

5 Check-ins prediction

As we mentioned above, we will build a predictive model for each cluster. Each cluster contains the places with similar attendance and every place has a history of check-ins changes. The inputs of the models will be the hour and the day of the week and the output – the number of check-ins at given time.

To build a predictive model, we will train a neural network using normalized check-ins data (popularity) and then transform the popularity back to the check-ins changes using a reversed version of the popularity formula, described above:

$$C = P * mean(C_{nz})$$

Where:

C - Number of check-ins
P - Popularity
C_{nz} - Object's all non-zero check-ins

In case predicted *C* is negative, it should be replaced with zero.

In this work a neural network with the resilient backpropagation training function (trainrp) is used for popularity prediction. The performance function of the network is mean square error (MSE). The same method to solve a similar problem was used in [10] work. The network has two neurons in the input layer, ten neurons in the single hidden layer and one neuron in the output layer.

The data of each cluster was divided into training and testing data sets. The training data set contains 60% of the cluster's data, other 40% are used for testing. The clusters with ten places or less were removed from this step due to a small amount of samples.

Table 5. Prediction errors.

Cluster	Training set size	Testing set size	Predicted and actual popularity MAE	Randomly predicted and actual popularity MAE	Maximum actual popularity
2	19577	13050	0.26	0.52	17.07
4	14053	9368	0.32	0.53	23.70
6	14523	9681	0.38	0.55	23.04
7	11602	7734	0.32	0.53	8.33
12	7938	5292	0.41	0.6	8.12
14	1848	1231	0.35	0.54	7.51
15	8379	5586	0.39	0.53	20.86
16	2536	1690	0.31	0.53	5.24
18	6248	4164	0.37	0.55	9.72
19	3509	2338	0.34	0.54	4.58
21	3220	2146	0.34	0.56	4.95
22	2103	1401	0.43	0.56	7.88
24	6975	4650	0.43	0.59	7.06
25	5955	3969	0.31	0.52	9.06

There are different metrics to evaluate predictive models: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), etc. [11]. In the given work to evaluate the performance of the predictive models we use Mean Absolute Error (MAE), which is easy to interpret and compute.

$$MAE = mean(|y - p|)$$

Where:

y - Actual value

p - Predicted value

To evaluate the models we calculated MAE of actual and predicted popularity values. The result is shown in Table 5. The average popularity MAE of all models is 0.35, while the average popularity of all places within the clusters is equal to one. The smallest error is within the cluster 2, where the MAE value is 0.26. The clusters 22 and 24 have the

highest error, which is 0.43. Also, to evaluate the models we compared the errors of our models and the model that outputs random popularity between zero and one (5th column in Table 5). The average error of random model is 0.55, which is 1.54 times bigger than the average error of our models.

To determine the impact of the clustering, we trained an overall model using the data of all places and compared its error with the average error of the separate models. The popularity MAE of the overall model is 0.57, which is 1.61 times larger than the average MAE of the separate models. It means that the clustering noticeably improves the prediction result.

6 Conclusion

Despite the fact that it is not possible to request historical visiting information of all the places due to the API limitations mentioned above, the presented approach shows good results on predicting attendance at different POIs. On the other hand, the amount of the POIs used is relatively small and it is possible that the system will not work so well on other areas. Also, the data used in clustering and prediction was collected during a small period of time. As mentioned previously, this was one of the reasons some places failed to share the cluster with some other objects as there was an unusual activity at these places at some day. Attendance at POIs could also vary depending on season, holidays, special event or even weather. Besides that, Foursquare and other similar social networks do not provide full information about visits, because not all the visitors use these types of social networks.

The given approach can be used in bigger tourism recommender systems to provide users better recommendations depending on user's preferences and desired period of time. Besides that, the given approach can be applied in other domains, for example, to predict attendance at special events and so on.

7 Related work

Popularity prediction and analysis of user behaviour attract more and more attention. These are the topics consumers as well as service providers are interested in, because it is helping providers to improve their services and optimize their resources and consumers to find the most suitable ones.

Similar work was done by T. Räsänen and M. Kolehmainen in [10]. The authors proposed an approach to predict regional visitor attendance levels for next seven days. As opposed to our work, the amount of features used in their work is much bigger. They used such information like regional weather conditions, mobile telecommunication event data, etc. Also, their work is fully based on neural networks (self-organizing map and multilayer perceptron), while we are using neural network only for prediction.

Another similar work was done by Jonathan Reades et al. in [5] paper. Having cellular activity data (Erlang information) of the area in Rome collected during four months, they showed how it is possible to normalize the data and did a cluster analysis. A similar technique was used in our work to cluster the POIs.

Xiaomei Zhang et al. in [12] offer a solution to predict an event attendance using event-based social network data. They trained and compared three models (logistic regression, decision tree and naive bayes) that predicted if a user will attend some future event basing on user's past visiting information. If we had an access to personal visiting information, we could use the approach proposed in the paper to predict the attendance at POIs.

[13] study the main features that make places popular and also the correlation between a place's category and a number of check-ins, a number of venues and visitor's loyalty. In another paper [14] an approach was made to find the correlations between Foursquare users' personalities and places they have visited. The main questions these studies answer is what makes POIs popular and by what types of users these POIs are mainly visited, while our work is concentrated on how popular POI is at desired time.

Popularity prediction is also topical in other fields, for example, web content popularity prediction. The authors of [15] proposed two popularity prediction models, which predict future view count of YouTube videos basing on information about number of views in the first days after video upload. These models showed significant improvement of

prediction accuracy compared to the model proposed in the earlier study [16]. Different web content popularity prediction models, features and evaluation metrics were discussed in [17] and [18]. These surveys also give a detailed overview of models' performance and challenging problems and suggest some applications of the models.

Different from these studies, we study the possibility of predicting popularity of the POIs having a limited access to the data.

8 Summary

The main purpose of this work is to develop an approach to predict attendance at different POIs using limited amount of data. To solve this kind of a problem we consider and apply different machine learning techniques. The whole process consists of data collection and preparation, cluster analysis of POIs and prediction model building.

The thesis is composed of six main chapters, each of the chapters describes different aspect of the approach. Chapter One is introductory and describes the background, goals and methodology of the work.

Chapter Two contains information about data source and available information. In this chapter we also discuss different data source limitations.

Chapter Three is divided into two parts. Part One describes the preparation process of the data, which is then examined in the second part.

Chapter Four concentrates on cluster analysis. The chapter begins with a description of data normalization and definition of popularity. Then, the k-mean clustering algorithm and distance function are discussed. Next, we define the attendance profile (a set of features) of the places and describe feature selection process. Finally, we determine the total number of clusters and do the clustering. In the end of the chapter we discuss the result and provide some recommendations on classification.

Chapter Five focuses on development of a check-ins prediction model. This chapter describes a predictive modelling process and provides the result, which is discussed in the end of the chapter.

Conclusion is drawn in Chapter Six, where we discuss the result of the work and different problems.

9 Future plans

In the given work the main responsibilities are divided between independent applications: data collecting application, data processing application and MATLAB. This prevents us from integrating it with some bigger recommender system, so the first step would be to implement a solid infosystem which will combine all mentioned responsibilities and also provide an access to the data and predictions for other applications. This infosystem also needs to be autonomous and adaptive, it must continuously collect the data, automatically cluster the POIs and rebuild predictive models.

Another goal is to expand observed territory and collect data about cities from all around the world. As already mentioned above, with this amount of the data the speed of the clustering and predictive model building processes will decrease. In this case clustering can be replaced with classification. A new place can be classified basing on the attendance profile to some existing cluster. And in case of prediction the offline method used in this work can be replaced with an online learning method.

Also, it is possible to improve the system by using data sources other than Foursquare as Foursquare does not provide historical information about check-ins. One more improvement is to collect and use information about special events, holidays and even weather. This information will help to provide better prediction, since the attendance at different POIs can vary depending on many other factors other than the hour of the day and the day of the week.

10 References

[1] "UNWTO Annual Report 2015," 2016. [Online]. Available: http://www2.unwto.org/publication/unwto-annual-report-2015.

[2] "The world's most tourism-dependent countries," 2013. [Online]. Available: https://skift.com/2013/03/09/the-worlds-most-tourism-dependent-countries/.

[3] "Foursquare," [Online]. Available: https://foursquare.com/about.

[4] "List of European cities by population within city limits," [Online]. Available: https://en.wikipedia.org/wiki/List_of_European_cities_by_population_within_city_limits.

[5] J. Reades, F. Calabrese, A. Sevtsuk and C. Ratti, "Cellular Census: Explorations in Urban Data Collection," 2007.

[6] "Euclidean and Euclidean Squared," [Online]. Available: http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm.

[7] "Variance Threshold," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html.

[8] T. Kodinariya and P. Makwana, "Review on determining number of Cluster in K-Means Clustering," 2013.

[9] "Distance weighted k-NN algorithm," [Online]. Available: http://www.data-machine.com/nmtutorial/distanceweightedknnalgorithm.htm.

[10] T. Räsänen and M. Kolehmainen, "Neural Network based method for predicting regional visitor attendance levels in recreational areas," 2008.

[11] "Evaluating forecast accuracy," [Online]. Available: https://www.otexts.org/fpp/2/5.

[12] X. Zhang, J. Zhao and G. Cao, "Who Will Attend? – Predicting Event Attendance in Event-Based Social Network".

[13] Y. Li, M. Steiner, L. Wang, Z.-L. Zhang and J. Bao, "Exploring Venue Popularity in Foursquare".

[14] M. J. Chorley, G. B. Colombo, S. M. Allen and R. M. Whitaker, "Visiting Patterns and Personality of Foursquare Users".

[15] H. Pinto, J. M. Almeida and M. A. Gonçalves, "Using early view patterns to predict the popularity of YouTube videos," 2013.

[16] G. Szabo and B. A. Huberman, "Predicting the Popularity of Online Content," 2008.

[17] A. Tatar, M. Dias de Amorim, S. Fdida and P. Antoniadis, "A survey on predicting the popularity of web content," 2014.

[18] S. Yu and S. Kak, "A Survey of Prediction Using Social Media," 2012.

Appendix 1 – Clusters

Cluster	Number of places	Categories (number of occurrences)	Cities (number of occurrences)
1	1	Hockey Arena (1)	Česká republika (1)
2	204	Neighborhood (21), Plaza (12), Café (10), Cocktail Bar (9), Metro Station (8), City (6), Hotel (6), Train Station (5), Bar (5), Train (5), Monument / Landmark (4), Art Museum (4), Concert Hall (4), Convention Center (4), Nightclub (4), Park (4), Soccer Stadium (3), Shopping Mall (3), Supermarket (3), Bridge (3), Pedestrian Plaza (3), Department Store (3), Italian Restaurant (2), Pub (2), Church (2), Bus Stop (2), Opera House (2), Fair (2), Farmers Market (2), Fountain (2), History Museum (2), Multiplex (2), Christmas Market (2), English Restaurant (1), Light Rail Station (1), Event Space (1), Music Venue (1), Gym / Fitness Center (1), Latin American Restaurant (1), Restaurant (1), Tram Station (1), Asian Restaurant (1), Bed & Breakfast (1), Fast Food Restaurant (1), Toy / Game Store (1), Beer Bar (1), Soccer Field (1), Tea Room (1), French Restaurant (1), Basketball Stadium (1), Field (1), Wine Bar (1), Bistro (1), High School (1), Nursery School (1), Office (1), Housing Development (1), Fried Chicken Joint (1), Road (1), Diner (1), Palace (1), Theater (1), Lounge (1), Hospital (1), Library (1), Art Gallery (1), Airport (1), General Entertainment (1), Historic Site (1), Dessert Shop (1), Country (1), Cultural Center (1), Electronics Store (1), Brewery (1), Comfort Food Restaurant (1), Pizza Place (1), Ice Cream Shop (1), Burger Joint (1), Roof Deck (1), Chinese Restaurant (1), Market (1), Vietnamese Restaurant (1), Castle (1),	United Kingdom (42), Česká republika (36), Ελλάδα (36), España (17), Nederland (13), Magyarország (12), Portugal (12), France (10), Deutschland (7), Italia (6), Österreich (5), Ireland (5), Eesti (3)

Cluster	Number of places	Categories (number of occurrences)	Cities (number of occurrences)
		Stadium (1), Spanish Restaurant (1), Mediterranean Restaurant (1)	
3	2	Soccer Stadium (1), City (1)	España (1), Nederland (1)
4	80	Metro Station (10), Plaza (10), Neighborhood (9), Shopping Mall (7), Train Station (6), Bus Station (4), Church (2), Café (2), Department Store (2), Road (2), American Restaurant (2), Gourmet Shop (2), Airport (2), Port (1), French Restaurant (1), Library (1), Sushi Restaurant (1), History Museum (1), Electronics Store (1), Turkish Restaurant (1), Indian Restaurant (1), Clothing Store (1), Capitol Building (1), Historic Site (1), Museum (1), Gym / Fitness Center (1), Tram Station (1), Bus Stop (1), College Library (1), Noodle House (1), Coffee Shop (1), Bridge (1), Tapas Restaurant (1)	Česká republika (19), United Kingdom (15), Magyarország (11), Deutschland (9), España (8), Österreich (7), Italia (5), Eesti (2), France (2), Portugal (1), Ireland (1)
5	1	Train Station (1)	Österreich (1)
6	56	Neighborhood (8), Train Station (6), Plaza (4), Church (4), Castle (3), Art Museum (3), Monument / Landmark (3), Historic Site (3), Park (2), Art Gallery (2), Palace (1), Bridge (1), Museum (1), Market (1), Flea Market (1), Building (1), History Museum (1), General Entertainment (1), Shopping Mall (1), Capitol Building (1), Memorial Site (1), Country (1), City (1), Outdoor Sculpture (1), Café (1), Brewery (1), Tram Station (1), Bus Station (1)	United Kingdom (20), Deutschland (14), Česká republika (8), Nederland (5), Italia (3), France (2), Vatican (2), España (1), Magyarország (1)
7	71	Neighborhood (7), Plaza (7), Café (6), Multiplex (5), Shopping Mall (4), Train Station (4), Concert Hall (3), Department Store (2), Park (2), Beer Bar (2), Music Venue (2), Austrian Restaurant (2), Movie Theater (1), Roof Deck (1), Coffee Shop (1), Chocolate Shop (1), Building (1), Indian Restaurant (1), Metro Station (1), Toy / Game Store (1), Stadium (1), River (1), Electronics Store (1), Garden (1), Opera House (1), Track (1), Road (1), Convention	Magyarország (15), United Kingdom (9), España (8), Deutschland (8), France (8), Česká republika (7), Italia (6), Eesti (4), Nederland (3), Österreich (3)

Cluster	Number of places	Categories (number of occurrences)	Cities (number of occurrences)
		Center (1), Event Space (1), Scenic Lookout (1), Marijuana Dispensary (1), City Hall (1), Performing Arts Venue (1), Italian Restaurant (1), Christmas Market (1), Beach (1), Church (1)	
8	1	Plaza (1)	España (1)
9	1	Plaza (1)	Česká republika (1)
10	1	Neighborhood (1)	United Kingdom (1)
11	1	Neighborhood (1)	United Kingdom (1)
12	31	Neighborhood (7), Plaza (7), Monument / Landmark (4), Department Store (2), City (2), Shopping Mall (2), Soccer Stadium (1), Bridge (1), Park (1), Road (1), Church (1), Fountain (1), Train Station (1)	United Kingdom (14), España (3), Deutschland (3), France (3), Magyarország (2), Nederland (2), Österreich (2), Italia (1), Česká republika (1)
13	1	Light Rail Station (1)	Deutschland (1)
14	18	Plaza (3), Multiplex (3), Neighborhood (2), Electronics Store (2), Light Rail Station (1), Church (1), Italian Restaurant (1), Beer Bar (1), Czech Restaurant (1), Tram Station (1), Supermarket (1), City (1)	Česká republika (9), Magyarország (3), Nederland (2), Deutschland (1), Österreich (1), Ireland (1), España (1)
15	39	Metro Station (13), Neighborhood (6), Train Station (5), Office (3), Plaza (3), University (3), Bus Station (1), Advertising Agency (1), Airport (1), Art Museum (1), Tunnel (1), Country (1)	United Kingdom (16), Česká republika (10), Magyarország (5), Deutschland (4), France (1), Vatican (1), Nederland (1), Österreich (1)
16	32	Multiplex (5), Plaza (4), Neighborhood (3), Movie Theater (2), Shopping Mall (2), Metro Station (2), Bar (2), Concert Hall (2), Food Court (1), Music Venue (1), Coffee Shop (1), Road (1), Theater (1), Tram Station (1), Pub (1), Indie Movie Theater (1), Department Store (1), Museum (1)	Česká republika (15), España (6), Magyarország (3), Eesti (2), Deutschland (2), Nederland (2), Italia (2)
17	1	French Restaurant (1)	United Kingdom (1)
18	34	Shopping Mall (8), Train Station (5), Plaza (5), Coffee Shop (2), Neighborhood (2), Gym / Fitness Center (2), Candy Store (1),	United Kingdom (9), Magyarország (9), Česká republika (9), France (3),

Cluster	Number of places	Categories (number of occurrences)	Cities (number of occurrences)
		Middle Eastern Restaurant (1), Pedestrian Plaza (1), Bridge (1), Church (1), Department Store (1), Furniture / Home Store (1), Pastry Shop (1), City (1), Light Rail Station (1)	Deutschland (1), Italia (1), Nederland (1), Österreich (1)
19	32	Shopping Mall (5), Plaza (4), Neighborhood (3), Department Store (2), City (2), Art Museum (1), Garden (1), Soccer Stadium (1), Park (1), Castle (1), Supermarket (1), Chocolate Shop (1), Market (1), Historic Site (1), Country (1), Café (1), American Restaurant (1), Food Court (1), Bridge (1), Harbor / Marina (1), Christmas Market (1)	España (7), France (6), Italia (5), Magyarország (4), Deutschland (3), Portugal (3), Česká republika (2), Österreich (1), Ελλάδα (1)
20	1	Shopping Mall (1)	Česká republika (1)
21	26	Neighborhood (3), Market (2), Monument / Landmark (2), Plaza (2), Art Museum (2), Department Store (2), City (2), Burger Joint (1), Electronics Store (1), Church (1), Road (1), Shopping Mall (1), American Restaurant (1), Christmas Market (1), Park (1), Waterfront (1), Airport (1), River (1)	Česká republika (7), Italia (4), United Kingdom (3), France (3), España (3), Nederland (3), Österreich (2), Magyarország (1)
22	11	Hotel (4), Greek Restaurant (1), Asian Restaurant (1), Pub (1), Metro Station (1), Café (1), Neighborhood (1), City (1)	United Kingdom (6), Ελλάδα (3), Magyarország (1), Česká republika (1)
23	3	Clothing Store (1), Market (1), Metro Station (1)	United Kingdom (2), Česká republika (1)
24	25	Train Station (15), City (5), Neighborhood (4), Airport (1)	United Kingdom (8), Deutschland (8), France (3), España (2), Italia (2), Magyarország (2)
25	53	Plaza (7), Shopping Mall (6), Park (3), Art Museum (3), Church (3), Train Station (2), Coffee Shop (2), Farmers Market (2), Capitol Building (1), Modern European Restaurant (1), History Museum (1), Science Museum (1), Café (1), Garden (1), Concert Hall (1), Opera House (1), Castle (1), Boat or Ferry (1), Pool (1), Sculpture Garden (1), Palace (1), Canal (1), Food	Magyarország (11), Česká republika (8), United Kingdom (7), España (7), Nederland (6), France (4), Österreich (4), Italia (3), Vatican (1), Ireland (1), Ελλάδα (1)

Cluster	Number of places	Categories (number of occurrences)	Cities (number of occurrences)
		Court (1), Electronics Store (1), Toy / Game Store (1), Restaurant (1), Road (1), Department Store (1), Art Gallery (1), City Hall (1), Cultural Center (1), Neighborhood (1), Monument / Landmark (1)	

Appendix 2 – Code repositories

Data collection application repository:

https://bitbucket.org/dmkaaa/checkin-tracker

MATLAB scripts:

https://bitbucket.org/dmkaaa/checkins_prediction