TALLINN UNIVERSITY OF TECHNOLOGY

Faculty of Information Technology

Anna Grund 179823IAIB

TOURISTIC BEHAVIOR RECOGNITION BASED ON *FOURSQUARE* DATA SETS

Bachelor's thesis

Supervisor: Priit Järv
PhD

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Anna Grund 179823IAIB

TURISTIDE KÄITUMISE TUVASTUS KASUTADES *FOURSQUARE* ANDMESTIKKU

Bakalaureusetöö

Juhendaja: Priit Järv
PhD

Tallinn 2020

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:	Anna Grund	....
		(signature)
Date:	18.05.2020	

Abstract

Human mobility data, produced in location-based social networks (LBSNs) enables us to study more about users' preferences, movement patterns and dynamics. Analysed data sets from LBSNs could also be used in urban planning and commercial purposes.

The thesis objective is to extract tourists behavior patterns by using individual histories of place visits from *Foursquare* [1] search-and-discover recommendation service. Besides tourists activities *Foursquare* data sets contain residents behavior which is not in our interest. By separating tourist activities from other activities we prepare the data for a tourist recommender system [2].

As a contribution, we developed mining algorithm and baseline algorithms for results validation. We also processed source data and annotated the data sets for potential future work purposes.

The thesis is written English and contains 52 pages of text, 6 chapters, 13 figures, 10 tables.

Annotatsioon

Asukohapõhistes sotsiaalvõrgustikes (LBSN) toodetud andmed inimeste liikuvuse kohta võimaldavad meil rohkem uurida kasutajate eelistuste, liikumisharjumuste ja dünaamika kohta. LBSNide analüüsitud andmestikke võiks kasutada ka linnaplaneerimisel ja ärilistel eesmärkidel.

Lõputöö eesmärgiks on saada turistide käitumismustreid, kasutades inimeste varasemaid külastusi *Foursquare'i* [1] otsingu ja avastamise soovitusteenuse andmetest. Lisaks turistide tegevustele sisaldavad *Foursquare'i* andmekogumid kohalike elanike käitumist, mis pole meie huvides. Eraldades turismitegevuse muudest tegevustest, valmistame andmed ette turistide soovitussüsteemi jaoks [2].

Oma panusena arendasime kaevandamise algoritmi ning algoritmi valideerivaid algoritme - alusjooni. Samuti töötlesime lähteandmeid ja koostasime anoteeritud andmekogumid potentsiaalse tulevase töö jaoks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 52 leheküljel, 6 peatükki, 13 joonist, 10 tabelit.

Table of Contents

List of Figures

List of Tables

List of abbreviations and terms

TalTech	Official abbreviation of Tallinn University of Technology
LBSN	Location-Based Social Network
HMM	Hidden Markov Model
MCMC	Markov Chain Monte Carlo
EM	Expectation–maximization
CF	Collaborative Filtering

1 Introduction

Human mobility data, produced in location-based social networks (LBSNs) enables us to study more about users' preferences, movement patterns and dynamics. Analysed data sets from LBSNs could also be used in urban planning and commercial purposes.

The thesis objective is to extract tourists behavior patterns by using individual histories of place visits from *Foursquare* [1] search-and-discover recommendation service. Besides tourists activities *Foursquare* data sets contain residents behavior which is not in our interest. By separating tourist activities from other activities we prepare the data for a tourist recommender system [2].

The expected output are check-ins from the data sets which are classified as tourist activities. We test three algorithms developed in the thesis - home location based, Hidden Markov Model and Extended Hidden Markov Model algorithms.

For results validation we implemented the baseline algorithms: dominating city algorithm, 30-day interval algorithm [3–5] and *The Maeda et al.* algorithm [2][6]. To the best of our knowledge, the first two mentioned are the main tourist extraction algorithms and *The Maeda et al.* is the newest of the baselines algorithm.

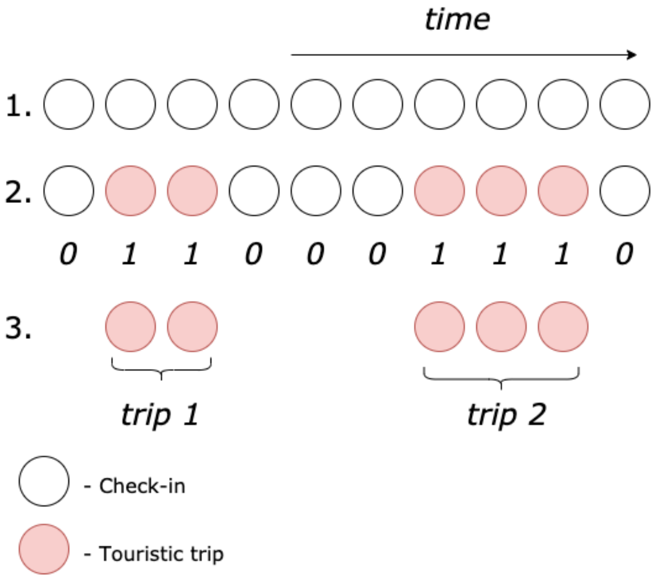


Figure 1. The process of tourists' trips extraction. 1 - having check-ins sets as input, 2 - potential trips detection, 3 - trips extraction

2 Related work

In this chapter we make an overview of the related works. These works either explore touristic behaviour patterns or directly use *Foursquare* data sets. However in contrast to our research, researches described below use different input data or if use *Foursquare* data sets, are used in different focuses and not validated [3, 4, 7–13].

We admit that during topic investigation, we revealed that there were methods invented to extract tourists and residents from *Foursquare* data sets, though the extraction itself was not in the focus of the work and the results were not validated [5][14].

In section 2.1 we make an overview of the existing researches on tourist behaviour extraction. In section 2.2 we make an overview of the existing researches using *Foursquare* data sets. We summarise analyses made on related work in 2.3, comparing the existing work to the contribution of the thesis.

2.1 Researches based on touristic behaviour extraction

In this section we make an overview of the existing researches on tourist behaviour extraction.

2.1.1 Automatic Construction of Travel Itineraries using Social Breadcrumbs

The goal of the paper is to automatically construct travel guidebooks by using shared photos (their geo-temporal locations, time path, contextual information) from *Flickr* [15] image hosting service [4]. By analyzing metadata of user's picture, it becomes possible to determine the cities visited by a person, duration of stay and transit time.

The research [4] faces several challenges as *Flickr* data sets' geo-location is not precise enough and location information might sometimes be misleading. In addition, some users tend to do posting more often than others and moving speed of different travel groups could strongly vary.

As a contribution, a better end-to-end analysis was made, implemented multiple heuristics pipelines and formed individual trips into graph representing touristic behaviour.

In contrast to our research, different data sets are being used and though analysis on tourists records is being performed, the final goal is not in patterns extraction but in construction of automated travel guidebook.

2.1.2 Leveraging explicitly disclosed location information to understand tourist dynamics: a case study

The goal of the research is to describe the approach to collect and analyse the history of physical presence of tourists from digital footprints they public on the web [3]. As a main resource image hosting service *Flickr* is used.

In order to separate residents from potential tourists the research uses developed algorithm. Taking 30-day periods, we compute the number of periods each photographer was active in the area. If all of the photos were taken within 30 days, we assume we deal with the tourist, otherwise we consider user to be a resident in the given time and area.

The paper's [3] method approach is mostly based on photos' metadata - as position and time of the last and first taken photo (in the area).

In contrast to our research, different data sets are being used. Additionally data for developed algorithm is not being produced but is fully based on metadata.

2.1.3 Friendship and Mobility: User Movement In Location-Based Social Networks

The goal of the research is to study three main aspects of human mobility: where do people move, how often do they move and what is the social impact on human movement [7]. The research [7] aims to point out what controls human mobility patterns and tries to estimate the likelihood of human movement to certain venues under certain conditions.

As input resource data from two social networks is used - *Gowalla* and *Brightkite*. In addition 2 million european mobile users are involved.

As a result of the research [7] Periodic & Social Mobility Model was developed for individuals mobility prediction. Developed model is based on three elements: model of spacial locations, model of temporal movement and model of movement inspired by social network.

In contrast to our research, different data sets are being used and the focus of the research is not in tourist behaviour extraction but in general in individuals movement tendencies.

2.1.4 Exploring the travel behaviors of inbound tourists to Hong Kong using geotagged photos

The goal of the research is to give an insight into travel behavior to better understand it and develop sustainable tourism industries [8]. The research [8] itself focuses on introduction of existing data mining techniques - density clustering and Markov Chain. As an input data from image hosting service *Flickr* is used.

Before using data in mining purposes metadata of user origin is being used to exclude all non-touristic activities in the region of Hong Kong.

The output of the research [8] are areas of tourist interests, popular touristic destinations within western and eastern tourists and tourists' moving trajectories and flows.

In contrast to our research, different data sets are being used, area of interest is very specific and the main focus is in discovering the locations in which tourists are interested the most and travel paths they take when travelling to Hong Kong.

2.2 Researches based on Foursquare data sets usage

In this section we make an overview of the existing researches using Foursquare data sets.

2.2.1 Detection of Spam Tipping Behaviour on Foursquare

The goal of the research stands in determination of tip spamming activities and related users and creation an automated tool for process detection in *Foursquare* search-and-discover recommendation service [9].

The research [9] uses *Foursquare* users' check-ins as input data.

The study [9] contribution is in distinguishing spam users activities and separating users with such activities from regular ones using machine learning techniques. Having separated all potential spam users characterization of irregular users behaviour is being performed. Spammers are being split into four categories: *Advertising/Marketing*, *Abusive*, *Self-promotion* and *Malicious*.

Though the study uses *Foursquare* data sets as input source, data is being used explicitly in detecting irregular users activities, classifying related users as spammers and splitting these into different categories.

2.2.2 An Empirical Study of Geographic User Activity Patterns in *Foursquare*

The goal of the research is to by using *Foursquare* platform users' check-ins analyse users' movement, geo-tagged dynamics and mobility patterns [10].

The research [10] uses *Foursquare* users' check-ins as input data both from *Foursquare* and *Twitter* [16] platforms.

During the study [10] the following aspects are covered: location-based marketing, locations correlations and general concept of user activity in certain time and place.

The output of the study [10] provides strong evidence about the power of LBSNs research opportunities. The study results could therefore be used in urban planning and social sciences.

Though the study [10] uses *Foursquare* data sets as input source, data is being used in general human movement and mobility studies.

2.2.3 You are where you eat: *Foursquare* checkins as indicators of human mobility and behaviour

The goal of the research is to analyse users' check-ins retrieved from *Foursquare* search-and-discover recommender platform [11]. The study [11] uses *Foursquare* check-ins collected during certain period of time. Specifically, the paper tends to investigate the differences between individuals using system to extract similar behaviour patterns.

The contribution of the research [11] stands in building the profiles of the mobility behaviour for platform users. These profiles might therefore be used in recommendation and suggestion systems.

Though the study [11] uses *Foursquare* data sets as input source, data is being used in general human movement and mobility studies for future use in recommendation and suggestion systems.

2.2.4 Recommendations in location-based social networks: a survey

The goal of the research is to make an overview of the existing recommender systems, which use different data sources and methods for recommendations making [12].

As we are interested in investigating the researches using *Foursquare* data sets we will focus on them notably.

The research [12] on *Foursquare* brings in power law distribution within *Foursquare* users' check-ins. Analysis makes it possible to better understand user and visiting venue correlations and patterns, find out users preferences for future use in recommender systems.

Methodologies described in the research [12] are categorized into three groups: content-based recommendation (uses user's profile), link analysis-based recommendation and collaborative filtering (CF) recommendation.

Some of the most used and significant techniques for making recommendations:

1. User profile - finding similarities between user's metadata (age, education, area of residence) and given location.

2. User location histories - considering online rating history of locations and ckeck-in history in LBSNs.

3. Filtering models - considering other users' online ratings, similarity inference calculation within users, recommendation score prediction.

4. Filtering models within friends only - it is considered to be as effective as finding similarities within top-k similar users.

5. User trajectories - user-generated movement trajectories are rich in its components: visit duration, order of venues visited, the path taken.

Though one of the analyses described in survey [12] uses *Foursquare* data sets as input source, data is being used in reccomender systems development with no prior tourist behaviour extraction. The research mainly focuses on learning human movement patterns and tendencies.

2.2.5 Beyond Sights: Large Scale Study of Tourists' Behavior Using Foursquare Data

The goal of the research is to analyse tourist behaviour using *Foursquare* data sets obtained from *Twitter* platform. Besides, the paper aims to estimate popular transitions among tourists and time when tourists visit certain venues [5].

The focus of the research [5] does not stand in tourist behaviour extraction, but in tourist behaviour and visited by tourists venues study. For tourists data sets separation users profiles metadata is used.

The study [5] of tourist behaviour could help to reveal most visited/popular venues within tourists and consequently build a better tourist recommender system. During the research a graph model with temporal attributes was developed to study spatio-temporal aspects of mobility patterns of tourists and residents.

Though the research [5] directly uses *Foursquare* data sets, the sets are used in tourist behaviour study and not extraction. Besides, within check-ins locations there are only four regions (London, New York, Rio de Janeiro and Tokyo) in the area of interest.

2.2.6 Investigation of Travel and Activity Patterns Using Location-based Social Network Data: A Case Study of Active Mobile Social Media Users

The goal of the paper is to investigate travel and activity patterns of *Foursquare* platform active users based on users' gender [14].

The research [14] mainly focuses on gender travel habits analyses as by investigating this aspect, it becomes possible to better understand the change in women's roles in family, labor force participation and society. In addition, data mined could be exploited to produce travel diary data.

For gender differences in the spatial density of activities and visited venues *Moran's I* statistic method was used to measure the autocorrelation of activities between male and female users.

As contributions of the work [14], additional knowledge on gender travel behaviour was gained (female users visit more distinct locations, male have longer trips and etc.).

Though the research [14] uses *Foursquare* data sets, the research topic stands in gender travel differences distinguishing. Also the area of interest is focused in New York City only.

2.2.7 Joint Modeling of User Check-in Behaviors for Point-of-Interest Recommendation

The aim of the research is to by building joint probabilistic generative model imitate user check-in behaviors during the decision making [13]. Additionally model aims to support two recommendation scenarios: home-town and out-of-town recommendations.

To achieve the goal, the method first estimates user's *home* location using metadata provided or by finding the location of the most made check-ins and considering it to be a *home* location.

The research [13] implements Markov Chain Monte Carlo (MCMC) probabilistic model with multiple parameters which impact on algorithm results (the activity range of user, the interests of user, the mean location and etc.)

As contributions of the work [13], a joint probabilistic model JIM was build and series of experiments on two recommendation scenarios were evaluated.

Though the research [13] uses *Foursquare* data sets and implements probabilistic model, the main focus is in modeling users' check-in behavior and solving the issue in data sparsity across geographical regions.

2.3 Summary

In this section we summarise analyses made on related work, comparing the existing work to the contribution of the thesis.

The aim of this chapter was not to report and simply analyse existing researches made on touristic behavior extraction or *Foursquare* data sets usage but to compare these researches with our thesis. With this analyses we aimed to emphasise that though the researches described above either explore touristic behaviour patterns or directly use Foursquare data sets, to the best of our knowledge, there were no researches made, which main focus was to extract touristic behaviour patterns and validate the results received.

The works described in section 2.1 use data sets from *Flickr* or *Brightkite+Gowalla* platforms and as a focus analyse visited by users places [3][8], construct probabilistic model for movement predictions [7] or aim to build algorithm for recommender system [4].

The works described in section 2.2 use data sets directly from *Foursquare* in various different focuses. Some focuses include determination of tip spamming activities within users [9], users' mobility and movement study [10] [11], user and visiting venue correlation study [12] and probabilistic model for prediction making construction [13].

During the thesis we discovered that there were researches made on tourist behaviour extraction [5][14], but the extraction itself was not the focus of the work and most importantly, the work results were not validated.

In the following table (Table 1) we summarise the results of the related work analyses more precisely described in sections 2.1 and 2.2.

Research	**Data source**	**Tourist extraction method**	**Tourist extraction validated**	**Focus/Topic**
Automatic Construction of Travel Itineraries Using Social Breadcrumbs [4]	*Flickr*	x-day interval	N	Recommender

Leveraging explicitly disclosed location information to understand tourist dynamics: a case study [3]	*Flickr*	x-day interval	N	Analyse visited places
Friendship and Mobility: User Movement in Location-Based Social Networks [7]	*Brightkite, Gowalla* (LBSN)	N	-	Probabilistic model to predict movement
Exploring the travel behaviors of inbound tourists to Hong Kong using geotagged photos [8]	*Flickr*	metadata	N	Analyse visited places
Detection of Spam Tipping Behaviour on Foursquare [9]	*Foursquare*	N	-	Determination of tip spamming activities
An Empirical Study of Geographic User Activity Patterns in Foursquare [10]	*Foursquare*	N	-	Users' movement, geotagged dynamics and mobility patterns
You are where you eat: Foursquare checkins as indicators of human mobility and behaviour [11]	*Foursquare*	N	-	General human movement and mobility studies
Recommendations in location-based social networks: A survey [12]	*Foursquare*	N	-	User and visiting venue correlations and patterns, human movement patterns and tendencies

Beyond Sights: Large Scale Study of Tourists' Behavior Using Foursquare Data [5]	*Foursquare*	Users profiles metadata	N	Tourist behaviour and visited by tourists venues study
Investigation of Travel and Activity Patterns Using Location-based Social Network Data: A Case Study of Active Mobile Social Media Users [14]	*Foursquare*	x-day interval	N	Gender differences in behavior
Joint Modeling of User Check-in Behaviors for Real-Time Point-of-Interest Recommendation [13]	*Foursquare*	N (inside model)	-	Probabilistic model to predict movement

Table 1. Related work summary

3 Methods and data

In this chapter we will introduce *Foursquare* data sets, methods developed during the thesis and implemented baselines.

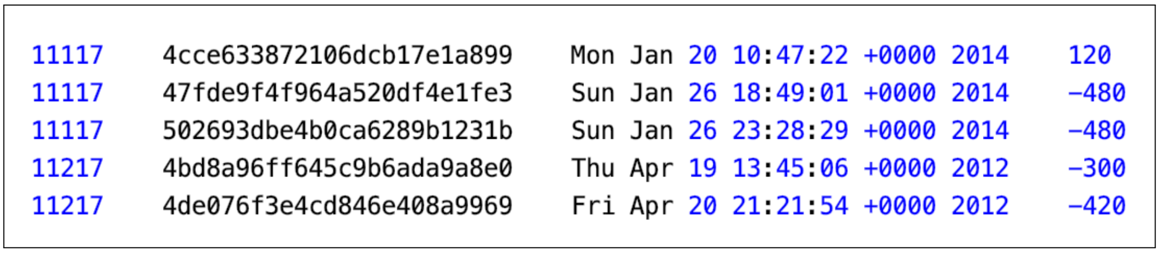
3.1 *Foursquare* data sets

As an input source we use data sets from *Foursquare* search-and-discover recommendation service. These data sets are rich in its amount and accuracy including 90,048,627 check-ins made by 2,733,324 users in 11,180,160 different venues during 03.04.2012 - 29.01.2014 period [17][18].

We are interested in using *Foursquare* data because we see multiple advantages in using these data sets. First of all, *Foursquare* data sets contain exact places (*venue id*) based on which we know exactly what was citizen's activity.

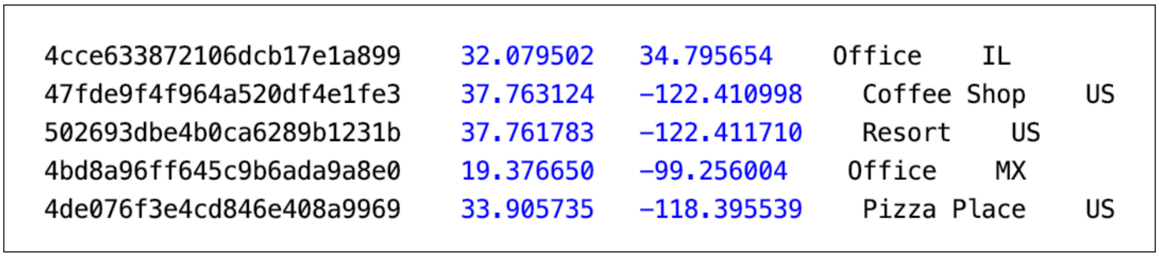
Secondly, besides *venue id*, we take advantage of exact geo location (latitude and longitude).

The data sets has originally two files - the check-ins (Figure 2) and the venue data (Figure 3).



```
11117    4cce633872106dcb17e1a899    Mon Jan 20 10:47:22 +0000 2014    120
11117    47fde9f4f964a520df4e1fe3    Sun Jan 26 18:49:01 +0000 2014    -480
11117    502693dbe4b0ca6289b1231b    Sun Jan 26 23:28:29 +0000 2014    -480
11217    4bd8a96ff645c9b6ada9a8e0    Thu Apr 19 13:45:06 +0000 2012    -300
11217    4de076f3e4cd846e408a9969    Fri Apr 20 21:21:54 +0000 2012    -420
```

Figure 2. Original data format from source: *user id, venue id, UTC time, timezone offset in minutes*

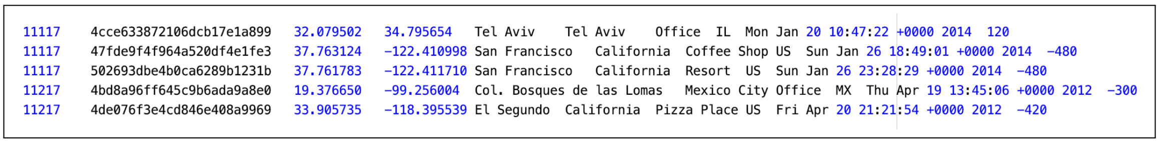


```
4cce633872106dcb17e1a899    32.079502   34.795654     Office    IL
47fde9f4f964a520df4e1fe3    37.763124   -122.410998     Coffee Shop    US
502693dbe4b0ca6289b1231b    37.761783   -122.411710     Resort     US
4bd8a96ff645c9b6ada9a8e0    19.376650   -99.256004     Office     MX
4de076f3e4cd846e408a9969    33.905735   -118.395539     Pizza Place     US
```

Figure 3. Original data format from source: *venue id, latitude, longitude, venue category name, country code*

We merge these data by venue id into one source. Additionally, we also resolve location using latitude and longitude. This enables us to mine data using area and city of check-in.

Final check-ins' version which is therefore being used for mining (Figure 4):



```
11117    4cce633872106dcb17e1a899   32.079502    34.795654   Tel Aviv    Tel Aviv    Office  IL  Mon Jan 20 10:47:22 +0000 2014  120
11117    47fde9f4f964a520df4e1fe3   37.763124    -122.410998 San Francisco   California  Coffee Shop US  Sun Jan 26 18:49:01 +0000 2014  -480
11117    502693dbe4b0ca6289b1231b   37.761783    -122.411710 San Francisco   California  Resort  US  Sun Jan 26 23:28:29 +0000 2014  -480
11217    4bd8a96ff645c9b6ada9a8e0   19.376650    -99.256004  Col. Bosques de las Lomas   Mexico City Office  MX  Thu Apr 19 13:45:06 +0000 2012  -300
11217    4de076f3e4cd846e408a9969   33.905735    -118.395539 El Segundo  California  Pizza Place US  Fri Apr 20 21:21:54 +0000 2012  -420
```

Figure 4. Merged data: *user id, venue id, latitude, longitude, venue category name, area, city, country code, UTC time, timezone offset in minutes*

The following scheme demonstrates *Foursquare* data sets preprocessing (Figure 5).

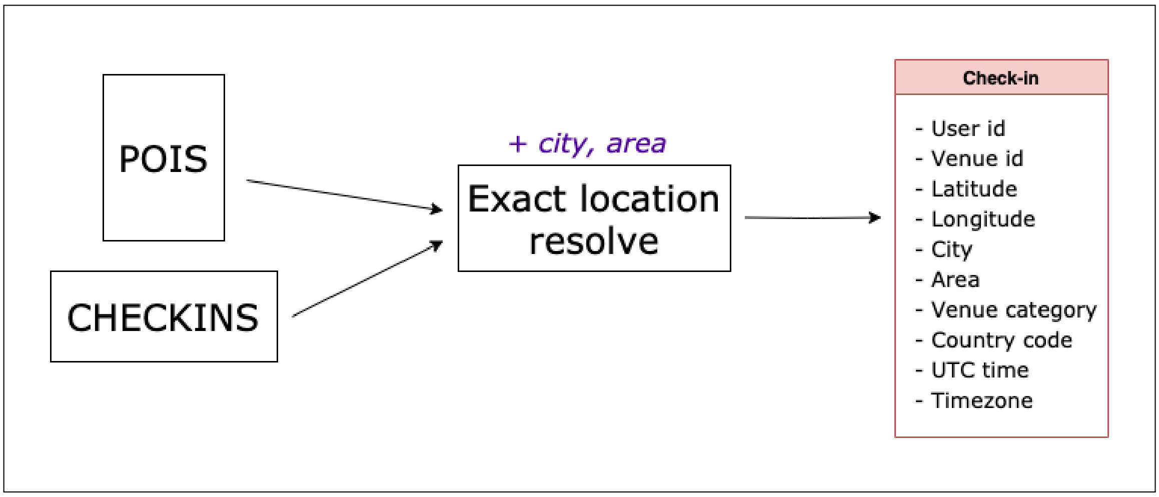


Figure 5. *Foursquare* data sets preprocessing

3.2 Technologies

During the thesis *Python* [19] as a main development language was used. Additionally we took advantage of some *Python* libraries as: *Reverse geocoder* [20], *Geopy* [21] and *Sklearn* [22] for better location resolve, *Numpy* [23], *Scipy* [24] - for better and quicker data computation and *Matplotlib* [25] for creating data visualisations.

For Markov Hidden Model implementation we used *Hmmlearn* [26] library.

3.3 Baselines

For the purposes of comparing developed algorithms performances we implemented three baseline algorithms: dominating locations algorithm, 30-day interval algorithm [3–5] and the *Maeda et al. algorithm* [2][6].

3.3.1 Dominating city algorithm

The main idea behind dominating city algorithm is for every user determining frequency for each city visit. The city with the largest number of visits for the user becomes his/her *home city*. All other locations are considered to be touristic locations.

Algorithm 1: Baseline 1: Dominating city algorithm

```
Input   : Check-ins $C$, users $U$
Output  : Check-ins labeled as tourist activities
foreach $u_i \in U$ do
    find $u_i$ user's check-ins $C$;
    foreach $c_j \in C_j$ do
        get each city visiting frequency;
    foreach $c_j \in C_j$ do
        if check-in's $c_j$ city = most frequently visited city then
            record $c_j$ as resident activity
        else
            record $c_j$ as touristic activity
```

3.3.2 30-day interval algorithm

The idea behind 30-day interval algorithm is described in one of the before mentioned related works [3–5]. In our thesis the algorithm is used as a baseline algorithm.

Taking 30-day periods, we compute the number of periods each user was active (had check-ins) in the area. If all of the check-ins were taken within 30 days, we assume we deal with a tourist, otherwise we consider user to be a resident in the given time and place.

Algorithm 2: Baseline 2: 30-day interval algorithm

```
Input  : Check-ins $C$, users $U$
Output : Check-ins labeled as tourist activities
foreach $u_i \in U$ do
    find $u_i$ user's check-ins $C$;
    foreach $c_j \in C_j$ do
        get check-ins visited by city;
    foreach city and $c_j$ visited check-ins do
        if |check-ins| = 1 then
            record $c_j$ as touristic activity
        else
            calculate days between $c_j$ and corresponding last check-in in city
            if difference in days > 30 then
                record $c_j$ as resident activity
            else
                record $c_j$ as touristic activity
```

3.3.3 The Maeda et al. algorithm

For the last baseline used in the thesis we have chosen density-based spatial clustering *Maeda et al.* algorithm [2][6].

The algorithm uses check-ins clustering for tourist behaviour extraction. Those user's check-ins which have neighbours in radius r in period of t unique days are assigned to a cluster (or reassigned to existing one). In our thesis the term *unique days* means how many days did user spend in radius r from some check-in, it gives us density, frequency about how often the user is in certain radius. These formed clusters are considered to be resident activities. Those check-ins not assigned into any cluster (meaning not having neighbours in radius r) might become potential touristic activities or noise. For each such check-in we search for unique users having the same or close check-in(s) in radius h from check-in. If unique users count exceeds p - check-in is recorded as touristic activity.

We analysed the algorithm giving different values for parameters. The highest results achieved with different parameters are presented in chapter 4.

```
Algorithm 3: Baseline 3: The Maeda et al. algorithm
Input      : Check-ins $C$, users $U$
Parameters : $t$ (days), $r$ (km), $p$ (density of check-ins), $h$ (density estimation
             bandwidth)
Output     : Check-ins labeled as tourist activities
foreach $u_i \in U$ do
    find $u_i$ user's check-ins $C$;
    foreach $c_j \in C_j$ do
        find neighbours $N_j$ of $c_j$ in radius $r$
        find unique days $T$ from the timestamps of $N_j$
        if $|T| >= t$ then
            assign check-ins $N_j$ to a new cluster
            if $N_j$ overlaps with existing clusters then
                re-assign their members to this cluster;
            else
        else
foreach $c_i \in C_i$ do
    if $c_i$ is not in a cluster then
        find unique users $U_i$ who have check-ins in radius $h$ from $c_i$;
        if $|U_i| >= p$ then
            record check-ins as touristic activity
        else
    else
```

3.4 Implementations

In this section we make an overview of the implemented solutions for extracting tourists behaviour patterns. During the thesis there were 3 algorithms implemented. These algorithms performance rates vary by parameters, options or additional checks we provide our algorithms with.

3.4.1 Hidden Markov Model

One of the algorithms implemented was using Hidden Markov Model (HMM). HMM is s a temporal probabilistic model where the state of the process is described by a single discrete random variable [27]. The possible values for the variable are the possible existing states of the world. We can split our world's states into two categories: *observable* and *hidden* states. Observable are these states which are seen.

Hidden on the opposite are those we cannot see nor observe and often should predict.

If we reflect it to our problem, observable states are those we see, specifically *user id, venue id, latitude, longitude, venue category name, area, city, country code, UTC time, timezone offset in minutes*. We are not however forced to use all of these for finding hidden states which in our work are *tourist* and *resident*.

In HMM we assume that each observable state's X_i behaviour depends on hidden state Z_i. By observing X_i we can learn about Z_i behaviour meaning that by observing user's U_i visible states we can learn (*predict*) U_i hidden states and conclude whether we deal with tourist or resident.

Following scheme demonstrates observable and hidden states relation (Figure 6).

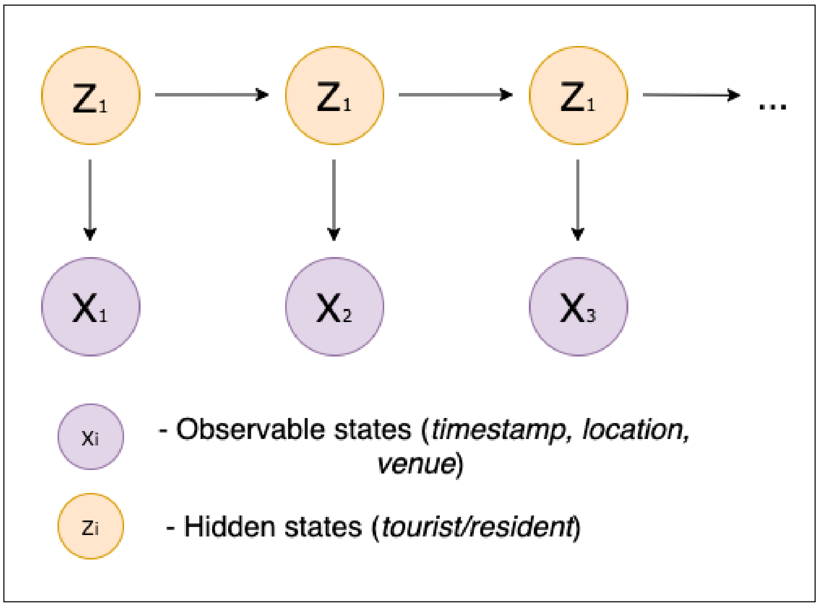
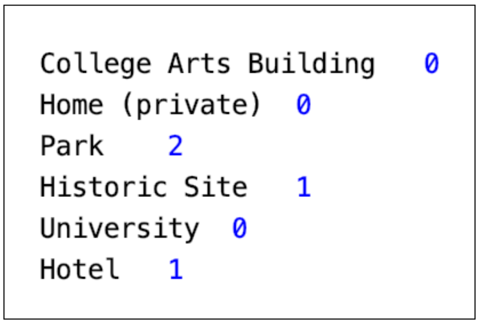

Z1
Z1
Z1
...
X1
X2
X3
Xi
- Observable states (*timestamp, location, venue*)
Zi
- Hidden states (*tourist/resident*)

Figure 6. Observable and hidden states relation

In our work we implemented HMM where observable state is *venue category* and hidden state is activity type (*resident/tourist*). We developed the algorithm in multiple versions: prediction algorithm (Viterbi/EM), probabilities provided/not provided. As probabilistic model we used multinomial distribution. The algorithms performing rates are presented in chapter 4.

```
Algorithm 4: Hidden Markov Model algorithm
Input      : Check-ins $C$, users $U$, start probabilities $Sp$, transition
             probabilities $Tp$, emission probabilities $Ep$, venue lookup table $Vt$
Parameters : Algorithm type $At$
Output     : Check-ins labeled as tourist activities
X = { }
foreach $u_i \in U$ do
    find $u_i$ user's check-ins $C$;
    foreach $c_j \in C_j$ do
        collect venue $v_j$ and transform it to state
predict hidden states based on venue $v$ states collected $(X)$ using $At$ algorithm;
```

In the algorithm presented above we use venue lookup table previously constructed using initial data source. We have analysed all of the possible venues categories and have given approximate, the closest possible state. The venues that more probably refer to resident we have labeled as "0", those representing touristic venues as "1" and those which exact state estimation is debatable as "2" (Figure 7).



```
College Arts Building   0
Home (private)   0
Park    2
Historic Site    1
University   0
Hotel    1
```

Figure 7. Venues lookup table example

Constructed lookup table is used for check-ins observable states resolve (on venue category) and following hidden states prediction.

We also provided our algorithm with the world start probabilities Sp, transition probabilities Tp and emission probabilities Ep. These probabilities were computed on manually labeled data sets.

For hidden states prediction we used methods in *Hmmlearn* [26] library specifying the algorithm for prediction process and including or excluding initial probabilities.

Hidden Markov Model based on venue: EM algorithm

For HMM implementation we used two different algorithms for predictions making. One of the algorithms used was EM. EM is a expectation–maximization algorithm which learns distribution parameters and the most probable hidden states [27].

Hidden Markov Model based on venue: Viterbi algorithm

Viterbi algorithm in contrast to EM algorithm assumes that distribution parameters are provided and aims to find the most probable hidden states [27].

Hidden Markov Model based on venue: parameters variations

As was mentioned before, in HMM development we not only vary with predictions algorithms but include or exclude the probabilities: start, transition and emission.

Start probabilities stand for general probability for check-in to have one or another state. In our model we assume, that for each check-in it is with 0.5 probability to have *resident* or *tourist* state (Table 2).

Resident	**Tourist**
0.5	0.5

Table 2. Model start probabilities

Transition probability informs us with what probability after state X_i will come state X_{i+1}. Precisely meaning with what probability after state *tourist* we will have state *resident* or *tourist.* These probabilities are computed using small subset of labeled data sets from the source (Table 3).

	Resident	**Tourist**
Resident	0.96	0.04
Tourist	0.34	0.66

Table 3. Model transition probabilities

In our model emission probability informs us with what probability while being tourist or resident will one have venue belonging to resident group ("0"), tourists group ("1") and undefined group ("2") (Figure 7). These probabilities are computed using small subset of labeled data sets from the source (Table 4).

Having once computed these probabilities on subset of 3026 out of 17 941 check-ins (106 of 500 users) we recorded them and use as true probabilities. We intentionally

	Resident venue ("0")	Tourist venue ("1")	Undefined venue ("2")
Resident	0.6008	0.0711	0.328
Tourist	0.6389	0.2947	0.0663

Table 4. Model emission probabilities

use the subset for probabilities computation to evaluate the model on entire labeled data sets (500 users).

However, it is possible to use our model without providing any of the probabilities. In this case the model generates these from observable states received. The differences in performance are presented in chapter 4.

3.4.2 Extended Hidden Markov Model

After developing plain HMM algorithm we decided to enhance existing model by preparing and additionally analyzing visible X_i states (venues categories). Partly, developed extension took an inspiration from previously described *Maeda et al.* algorithm where clustering pattern was used.

Algorithm 5: Extended Hidden Markov Model algorithm

```
Input      : Check-ins $C$, users $U$, $t$ (days), $r$ (km), start probabilities $Sp$,
             transition probabilities $Tp$, emission probabilities $Ep$, venue
             lookup table $Vt$
Parameters : Algorithm type $At$
Output     : Check-ins labeled as tourist activities
X = { }
foreach $u_i \in U$ do
    find $u_i$ user's check-ins $C$;
    foreach $c_j \in C_j$ do
        find neighbours $N_j$ of $c_j$ in radius $r$
        if |neighbours $N_j$| >= 1 then
            find difference in check-ins days between neighbours $N_j$
            if difference in check-ins days <= $t$ then
                $c_j$ state = 1
            else
                $c_j$ state = 0
        else
            if $Vt$ state = 0 then
                $c_j$ state = 0
            else
                $c_j$ state = 1
    collect states to $X$
predict hidden states based on states collected ($X$) using $At$ algorithm;
```

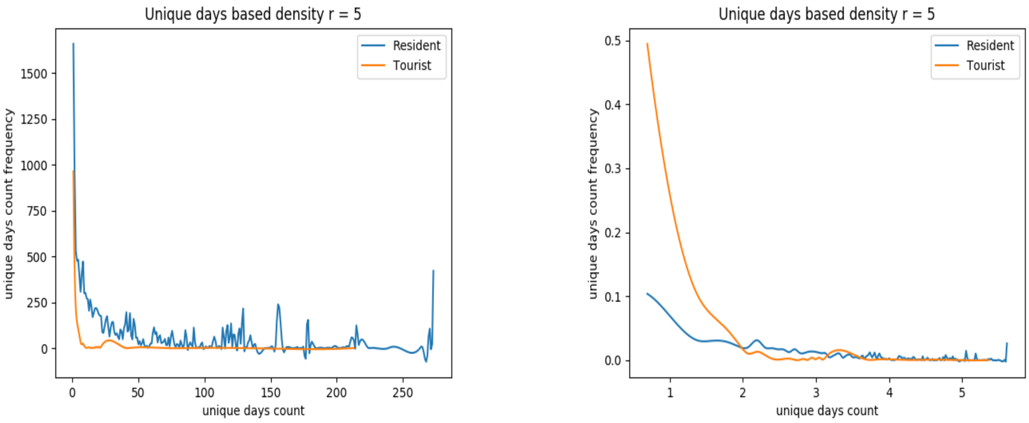
In algorithm described above before performing hidden states predictions, we analyse and aim to make "better visible states". We still use venues categories lookup table but in addition for each user's check-in we find neighbours in radius r and difference in check-ins days between these neighbours. If we find such neighbours that fit parameters r and t we assume that current check-in might be a part of touristic activity. In this case we label the check-in as *tourist*. However, if there are no neighbours or observable check-in does not fit in r and t we turn to lookup table.

We find the extension developed important as before prediction making, we make sure that observable states provide us with the complete information about the check-ins. We take into consideration neighbours (their presence) and the difference in days of neighbours check-ins.

Similarly to the plain HMM implementation, we specify the algorithm for prediction process and include or exclude initial probabilities.

In the theses for both HMM implementations we used multinomial distribution. However, after testing and visualising the extended algorithm, some new circumstances have appeared.

The following graphs (Figure 8) visualise unique days frequency for tourists and residents. In other words, here we observe how many unique days of travel each group has. When analysing the logarithmic graph representation, we conclude that tourists have much more unique days of travel (lasting for about 1 day) when residents' unique days distribution is more smooth. This model of distribution resembles a power law (*Pareto*) distribution, where the change in one group leads to proportional relative change in another group [28].



(a) Standard

(b) Logarithmic

Figure 8. Unique days density for Extended Hidden Markov Model

Power law distribution is relatively rare and identifying it might be challenging.

Looking at the results received, we believe that for future work it could be worth making effort to implement power law distribution in HMM algorithm.

3.4.3 Developed algorithm

Lastly, we have developed our own algorithm for tourist behaviour extraction.

Algorithm 6: Developed algorithm

```
Input  : Check-ins $C$, users $U$
Output : Check-ins labeled as tourist activities
foreach $u_i \in U$ do
    find the user's check-ins $C$
    foreach $c_j \in C_j$ do
        find potential home cities
    foreach $c_j \in$ home city checkins do
        find max distance between home cities
    foreach $c_i \in C_j$ do
        foreach $c_j \in$ home city checkins do
            if $c_i$ area = $c_j$ area then
                if distance between $c_i$ and $c_j$ <= max distance between home or
                  $c_j$ venue is specific then
                    record check-in as resident activity
                else
                    record check-in as touristic activity
            else
                record check-in as touristic activity
```

For each user's check-ins we first aim to estimate potential home cities (these might be multiple). We reveal the cities by finding venues which could possibly belong to *resident* group users such us *Home (private)*, *Office*, *Courthouse* and etc. If we cannot resolve home city (we do not achieve venues match), we treat all of the check-ins as home. Doing so might be quite rough because we exclude the possibility to find touristic check-ins (if we do not find any home city), but here we proceeded from the principle that we would better not make a recommendation *versus* we make it and it will be out of place.

After resolving home locations we estimate average coordinates and maximum distance between home locations. Additionally we also verify home cities' suburbs and label them as resident activities if the area of city matches home city area, their distance from average location is not greater than maximum distance between home locations or we deal with specific venue (*Airport*, *Road* and etc) in the area.

Consequently, all check-ins left, meaning not being added to *home* group, are labeled as touristic activities. The performance rate of the algorithm can be observed in chapter 4.

4 Results

In this chapter we bring up all of the results and findings achieved during the thesis. We also look into validation process details.

4.1 Validation

For algorithm performance rate comparison we use baseline algorithms precisely described in chapter 3.

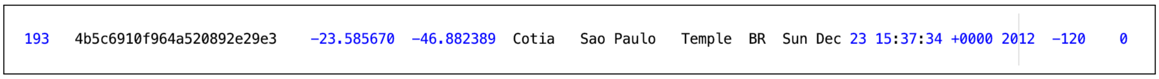
For results validation we use check-ins' manual labeling.

4.1.1 Manual labeling

In order to be able to accurately validate the results achieved we priorly label source check-ins as tourist or resident. True labeled data also supports us in HMM algorithm development (start, emission and transition probabilities calculations done on subset).

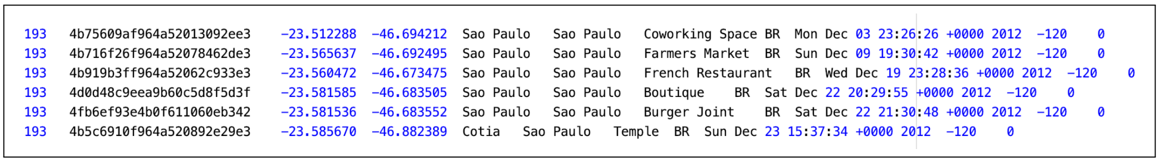
During the thesis we have analysed and manually labeled 17 941 random samples of check-ins belonging to 500 different users.

We analyse each user check-ins' set aiming to clarify what is the home location for user and what could be potential touristic activities. If we find that observing check-in belongs to "resident" definition, we label such as "0" (Figures 9 and 10).



```
193   4b5c6910f964a520892e29e3    -23.585670  -46.882389  Cotia    Sao Paulo    Temple  BR  Sun Dec 23 15:37:34 +0000 2012  -120    0
```

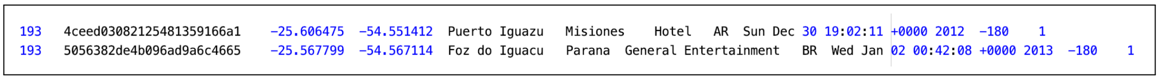
Figure 9. Check-in labeled as resident



```
193   4b75609af964a52013092ee3    -23.512288  -46.694212  Sao Paulo    Sao Paulo    Coworking Space BR  Mon Dec 03 23:26:26 +0000 2012  -120    0
193   4b716f26f964a52078462de3    -23.565637  -46.692495  Sao Paulo    Sao Paulo    Farmers Market  BR  Sun Dec 09 19:30:42 +0000 2012  -120    0
193   4b919b3ff964a52062c933e3    -23.560472  -46.673475  Sao Paulo    Sao Paulo    French Restaurant   BR  Wed Dec 19 23:28:36 +0000 2012  -120    0
193   4d0d48c9eea9b60c5d8f5d3f    -23.581585  -46.683505  Sao Paulo    Sao Paulo    Boutique    BR  Sat Dec 22 20:29:55 +0000 2012  -120    0
193   4fb6ef93e4b0f611060eb342    -23.581536  -46.683552  Sao Paulo    Sao Paulo    Burger Joint    BR  Sat Dec 22 21:30:48 +0000 2012  -120    0
193   4b5c6910f964a520892e29e3    -23.585670  -46.882389  Cotia    Sao Paulo    Temple  BR  Sun Dec 23 15:37:34 +0000 2012  -120    0
```

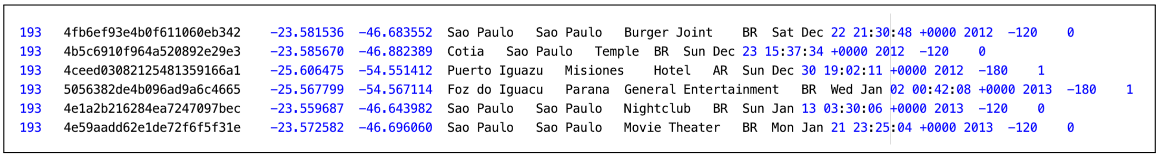
Figure 10. Group of check-ins labeled as residents

If after finding home location or location which seems to be in a sense permanent, we find check-in (check-ins) which location, duration and venue visited could potentially be touristic activities, we label this check-in as "1" (Figure 11).



```
193   4ceed03082125481359166a1    -25.606475  -54.551412  Puerto Iguazu   Misiones    Hotel   AR  Sun Dec 30 19:02:11 +0000 2012  -180    1
193   5056382de4b096ad9a6c4665    -25.567799  -54.567114  Foz do Iguacu   Parana  General Entertainment   BR  Wed Jan 02 00:42:08 +0000 2013  -180    1
```

Figure 11. Group of check-ins labeled as tourists



```
193   4fb6ef93e4b0f611060eb342    -23.581536  -46.683552  Sao Paulo   Sao Paulo   Burger Joint    BR  Sat Dec 22 21:30:48 +0000 2012  -120    0
193   4b5c6910f964a520892e29e3    -23.585670  -46.882389  Cotia   Sao Paulo   Temple  BR  Sun Dec 23 15:37:34 +0000 2012  -120    0
193   4ceed03082125481359166a1    -25.606475  -54.551412  Puerto Iguazu   Misiones    Hotel   AR  Sun Dec 30 19:02:11 +0000 2012  -180    1
193   5056382de4b096ad9a6c4665    -25.567799  -54.567114  Foz do Iguacu   Parana  General Entertainment   BR  Wed Jan 02 00:42:08 +0000 2013  -180    1
193   4e1a2b216284ea7247097bec    -23.559687  -46.643982  Sao Paulo   Sao Paulo   Nightclub   BR  Sun Jan 13 03:30:06 +0000 2013  -120    0
193   4e59aadd62e1de72f6f5f31e    -23.572582  -46.696060  Sao Paulo   Sao Paulo   Movie Theater   BR  Mon Jan 21 23:25:04 +0000 2013  -120    0
```

Figure 12. Group of check-ins labeled as residents and tourists

We assume one of a user can behave as tourist or as resident in given place and time.

For better and exact labeling we developed regulations list for tourist and resident labeling process.

We take into area of interest data sets of check-ins belonging to one user:

1. First, we try to define what is user's home location. Possible home location might be region where user has majority of check-ins. However, it is not always the case. We try to prove that current location might be treated as home by finding venues such as: *university*, *school*, *daycare*, *office*, *hospital* and etc.

2. We also consider the situations when user can stay and live in multiple cities. These cities and all activities taking place there are treated as residential activities. For this reason we analyse the whole user's data set.

3. We also consider cases where a person is on his/her work trip by searching for a *hotel* and *office*, *meeting room*, *startup tech* at one time and place. In this situation we do not consider these check-ins as touristic activities.

4. If there is an *airport* within extracted home locations - all other airports will point to touristic activities (trip) unless there is something to prove contrary (second home, work trip etc.).

5. There also exists a possibility that person can be a tourist in his/her own region (as for instance going to ski resort).

6. If we are to find *building*, *private home* and *hotel* at one location and time we assume we deal with a resident.

7. A check-in between two tourist's check-ins that occur on the same day is very likely part of the same touristic trip.

8. All single check-ins are considered to be touristic activities, unless they refer to *university*, *school*, *daycare*, *office*, *hospital* and etc.

These are key regulations for decision making process, however it is worth noticing that some check-ins labeling might come intuitively and the labeling results might be subjective.

After executing one of the algorithms developed or baselines we compare results received and expected results. For estimating algorithm's performance we use accuracy, recall, precision and F1 score (harmonic mean) as indicators of success.

4.2 Thesis outcomes

In this section we will make an overview of the results achieved during the thesis. We have implemented and validated 6 algorithms with different variations (parameters value, additional checks, conditions) (Table 5, Figure 13).

Algorithm	**Accuracy**	**Recall**	**Precision**	**F1 Score**
B 1: Dominating city algorithm	47.44	4.66	1.19	0.95
B 2: 30-day interval algorithm	75.48	46.39	21.26	29.16
B 3: Maeda et algorithm (T = 4, r = 15, p = 0)	85.45	**72.89**	40.61	**52.16**
Markov plain probabilities given (EM)	**87.56**	15.02	33.87	20.82
Extended Markov probabilities given (EM)	81.91	66.29	33.33	44.36
Developed algorithm	86.34	59.1	**41.11**	48.49

Table 5. Implemented algorithms' performance rates

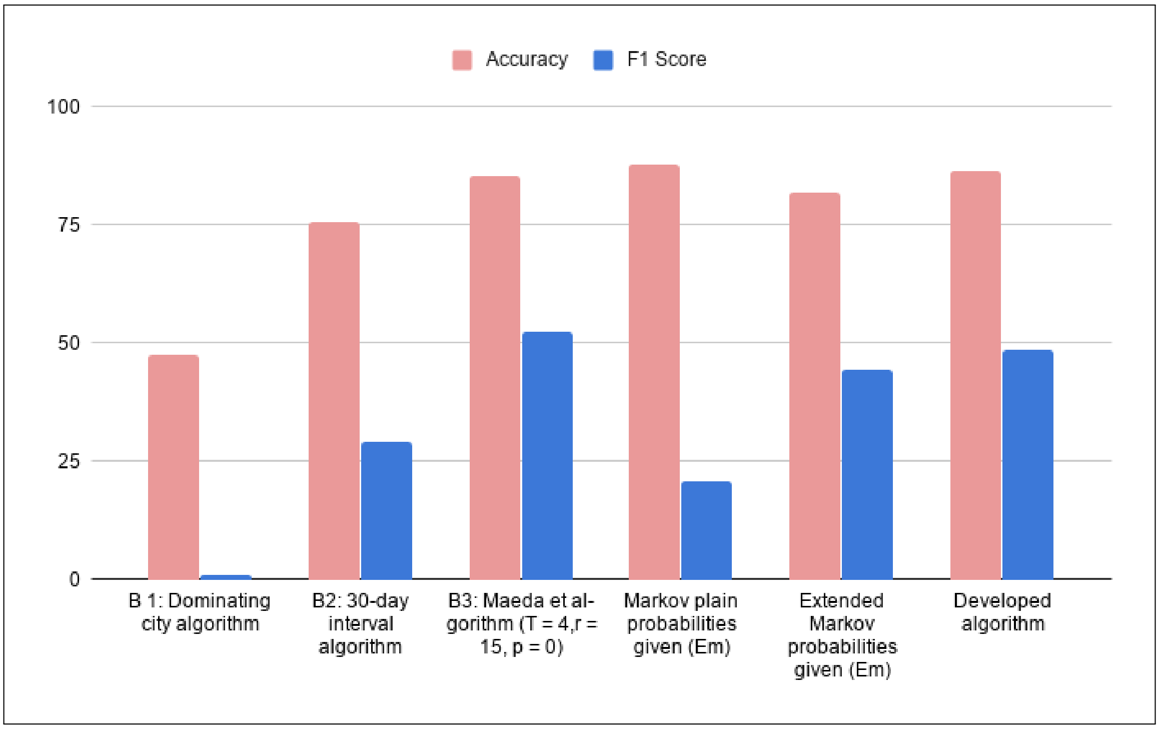


Figure 13. Implemented algorithms rating

When analysing the results we can conclude that the algorithm providing the most accurate performance rate is HMM (given probabilities and using EM algorithm for prediction making).

Accuracy provides us with classification reliability for all data sets. However, in the thesis we do not find accuracy to be very informative as tourists check-ins are in minority. For example during the thesis, having analysed 17 941 users check-ins we got 1952 check-ins of tourist behavior which makes only 10.88% of all check-ins.

For better clarity, we present accuracy equation, where *TP* stands for *true positive* (those check-ins that were supposed to be touristic and that actually were), *TN* - *true negative* (those check-ins that were supposed to be touristic but actually were not), *FP* - *false positive* (those check-ins that were not supposed to be touristic but actually were) and *FN* - *false negative* (those check-ins that were not supposed to be touristic and that actually were not) (Equation 4.1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

For this reason we are more interested in rating both recall and precision numbers. For their synchronous estimation we use F1 score (Equation 4.2, Figure 13).

$$F1 = \frac{recall \cdot precision}{recall + precision} \cdot 2 \tag{4.2}$$

The algorithm with the highest recall rate (how many of the correctly estimated touristic activities check-ins are selected) is *Maeda et al.* baseline algorithm having radius r as 15, unique days count T as 4 and unique users p as 0 (Equation 4.3). Also the baseline scored the highest F1 score (harmonic mean of precision and recall).

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

The algorithm having the highest precision (how many of the selected touristic check-ins are relevant, estimated correctly) is developed during the thesis algorithm (Equation 4.4).

$$Precision = \frac{TP}{TP + FP} \tag{4.4}$$

Proceeding from the results and the performance numbers we can claim that these is still space and motivation for future development and investigation. We see potential in both developed and Extended HMM algorithms and as a future work we see the need in implementing power low (*Pareto*) distribution and enhancing self-invented algorithm.

4.2.1 Algorithms' parameters

In the thesis, some of the algorithms implemented included parameters usage. For example in *Maeda et al.* algorithm there were radius r, unique days count T and unique users p. We tried r, T, p in different combinations and measured the algorithm performance rate. Also, in HMM implementations we included or excluded the model's probabilities.

In this section we will describe parameters impact on algorithms' performance.

Maeda et al. algorithm

When implementing *Maeda et al.* algorithm there was three main parameters impacting on algorithms performance rate - radius r, unique days count T and unique users p. In chapter 6 we have included the table of parameters variations.

Within all attempts made we revealed two *strongest* parameters configurations providing the highest performance scores.

The first parameters choice presented in Table 6 needs at least four unique days of check-ins having distance not more than 15 kilometers. All those check-ins left outside of clusters are automatically treated as touristic activities (because of the parameter p being equal to 0). This configuration brought up the highest recall and harmonic mean rates among all other implemented algorithms.

On the other hand, the second configuration showed much lower rate in recall and consequently in F1 score (for the reason that not every outside the cluster check-in is treated as touristic activity - p being equal to 1). However, there was slight improvement in accuracy and precision rates.

Algorithm	Accuracy	Recall	Precision	F1 Score
Maeda et algorithm (T = 4, r = 15, p = 0)	85.45	**72.89**	40.61	**52.16**
B 3: Maeda et algorithm (T = 4, r = 5, p = 1)	**88.41**	16.15	**41.56**	23.26

Table 6. *Maeda et al.* algorithm results

Since we deal with density based algorithm, the components amount (*check-ins*) becomes significant. The less check-ins (samples) we provide the baseline with, the lower is the probability that another user from the sample has visited the same venue. For this reason, we assume that the more check-ins we give as an input for the algorithm the more accurate results we receive. In Table 7 we demonstrate *Maeda et al.* algorithm on 35 243 check-ins within 1000 users (500 labeled and validated).

Algorithm	Accuracy	Recall	Precision	F1 Score
Maeda et algorithm (T = 4, r = 15, p = 0)	85.45	**72.89**	40.61	**52.16**
B 3: Maeda et algorithm (T = 4, r = 5, p = 1)	**87.06**	35.78	**39.52**	37.56

Table 7. *Maeda et al.* algorithm results (35 243 check-ins)

Based on the results received we can conclude that when using larger data sets we improve the accuracy, recall rate and consequently F1 score.

Hidden Markov Model

When implementing HMM algorithm we had two variations to choose from - probabilities inclusion or exclusion and predictions algorithm choice - EM or Viterbi. The deeper insight about model's parameters is described in subsection 3.4.1.

Here we did not take into consideration HMM using Viterbi algorithm and being not given probabilities. Viterbi algorithm does not include automated estimation of distribution and the results received might be misleading.

Algorithm	Accuracy	Recall	Precision	F1 Score
Markov plain probabilities given (Viterbi)	**84.2**	17.91	**40.51**	24.84
Markov plain probabilities given (EM)	83.64	19.27	37.95	25.56
Markov plain no probabilities (EM)	42.99	**92.06**	19.37	**32.0**

Table 8. Hidden Markov Model algorithm results

Extended Hidden Markov Model

Similarly to previously described approach, the results of Extended HMM are presented in Table 9.

Algorithm	Accuracy	Recall	Precision	F1 Score
Extended Markov probabilities given (Viterbi)	81.95	58.89	32.06	41.51
Extended Markov probabilities given (EM)	81.91	**66.29**	33.33	**44.36**
Extended Markov no probabilities (EM)	**85.14**	48.26	**36.26**	41.41

Table 9. Extended Hidden Markov Model algorithm results

5 Future work

Having achieved certain results described in chapter 4 within limed time frame we conclude that there is still space and motivation to investigate current topic. Based on the results received we assume there is a potential in implementing power law (*Pareto*) distribution for Extended HMM algorithm and enhancing developed during the thesis algorithm.

One subject that could also be explored is the amount of touristic behavior data which is possible to extract by using *Foursqure* data sets. For example during the thesis, having analysed 17 941 users check-ins we got 1952 check-ins of tourist behavior (10.88%).

6 Conclusions

The goal of the thesis was to extract tourists behavior patterns by using individual histories of place visits from *Foursquare* [1] search-and-discover recommendation service. For achieving this goal we developed three mining algorithms: Hidden Markov Model, Extended Hidden Markov Model and self-invented developed algorithm. For our algorithms validation we have also implemented three baseline algorithms: Dominating city algorithm, 30-day interval algorithm [3–5] and *Maeda et al.* [2][6] algorithms.

As a result of the thesis we conclude that the strongest algorithm implemented is *Maeda et al. (T = 4,r = 15, p = 0)* baseline gaining the best harmonic mean score (52.16). As a second conclusion we claim that previously unvalidated baseline 2 (30-day interval algorithm) turned out to be relatively weak and as a consequence could not be used as accurate tourist behavior extraction method. Coming back to the developed algorithm we conclude that the numbers achieved are comparatively high and are close to the maximum gained during the thesis (F1: 48.49), however, we see that F1 score barely exceeds half of the possible rate, which is still low and unconvincing. The set goal appeared to be more difficult objective than was estimated and in limited time frame could not be fully solved. As a last contribution we bring up the labeled *Foursquare* data sets (17 941 random samples of check-ins belonging to 500 different users) as it is valuable input mining data for any future work.

Though we believe, we have achieved some good performance numbers during the thesis, it is still not enough to treat this result as final. We see potential in improving developed during the thesis algorithm and implementing another distribution to Extended HMM (chapter 5).

Bibliography

[1] *Foursquare*. URL: `https://foursquare.com/city-guide`. (accessed: 01.03.2020).

[2] Priit Järv. “Mining Tourist Behavior from Foursquare Check-ins”. In: Tallinn, Estonia, 2019, pp. 1–6. URL: `http://sightsmap.com/misc/fsq_report.pdf`.

[3] Fabien Girardin et al. “Leveraging explicitly disclosed location information to understand tourist dynamics: a case study”. In: *Journal of Location Based Services* 2.1 (2008), pp. 41–56. DOI: `10.1080/17489720802261138`. eprint: `https://doi.org/10.1080/17489720802261138`. URL: `https://doi.org/10.1080/17489720802261138`.

[4] Munmun De Choudhury et al. “Automatic Construction of Travel Itineraries Using Social Breadcrumbs”. In: *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*. HT ’10. Toronto, Ontario, Canada: Association for Computing Machinery, 2010, pp. 35–44. ISBN: 9781450300414. DOI: `10.1145/1810617.1810626`. URL: `https://doi.org/10.1145/1810617.1810626`.

[5] A. P. G. Ferreira, T. H. Silva, and A. A. F. Loureiro. “Beyond Sights: Large Scale Study of Tourists’ Behavior Using Foursquare Data”. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015, pp. 1117–1124.

[6] Takashi Nicholas Maeda et al. “Extraction of Tourist Destinations and Comparative Analysis of Preferences Between Foreign Tourists and Domestic Tourists on the Basis of Geotagged Social Media Data”. In: *ISPRS Int. J. Geo-Information* 7.3 (2018), p. 99. DOI: `10.3390/ijgi7030099`. URL: `https://doi.org/10.3390/ijgi7030099`.

[7] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. “Friendship and Mobility: User Movement in Location-Based Social Networks”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’11. San Diego, California, USA: Association for Computing Machinery, 2011, pp. 1082–1090. ISBN: 9781450308137. DOI: `10.1145/2020408.2020579`. URL: `https://doi.org/10.1145/2020408.2020579`.

[8] Huy Quan Vu et al. “Exploring the travel behaviors of inbound tourists to Hong Kong using geotagged photos”. In: *Tourism Management* 46 (2015), pp. 222–232. ISSN: 0261-5177. DOI: https://doi.org/10.1016/j.tourman.2014.07.003. URL: http://www.sciencedirect.com/science/article/pii/S0261517714001356.

[9] Anupama Aggarwal, Jussara Almeida, and Ponnurangam Kumaraguru. “Detection of Spam Tipping Behaviour on Foursquare”. In: *Proceedings of the 22nd International Conference on World Wide Web*. WWW ’13 Companion. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, pp. 641–648. ISBN: 9781450320382. DOI: 10.1145/2487788.2488015. URL: https://doi.org/10.1145/2487788.2488015.

[10] Anastasios Noulas et al. “An Empirical Study of Geographic User Activity Patterns in Foursquare”. In: *ICWSM*. 2011. URL: https://www.semanticscholar.org/paper/An-Empirical-Study-of-Geographic-User-Activity-in-Noulas-Scellato/fd59f25da4eac25875c9f2f431970077e12a7d9c.

[11] G. B. Colombo et al. “You are where you eat: Foursquare checkins as indicators of human mobility and behaviour”. In: *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. 2012, pp. 217–222.

[12] Jie Bao et al. “Recommendations in location-based social networks: A survey”. In: *GeoInformatica* 19 (July 2015). DOI: 10.1007/s10707-014-0220-8.

[13] Hongzhi Yin et al. “Joint Modeling of User Check-in Behaviors for Real-Time Point-of-Interest Recommendation”. In: *ACM Trans. Inf. Syst.* 35.2 (Oct. 2016). ISSN: 1046-8188. DOI: 10.1145/2873055. URL: https://doi.org/10.1145/2873055.

[14] Yeran Sun and Ming Li. “Investigation of Travel and Activity Patterns Using Location-based Social Network Data: A Case Study of Active Mobile Social Media Users”. In: *ISPRS International Journal of Geo-Information* 4 (Aug. 2015), pp. 1512–1529. DOI: 10.3390/ijgi4031512.

[15] *Flickr*. URL: https://www.flickr.com. (accessed: 01.03.2020).

[16] *Twitter*. URL: https://twitter.com/explore. (accessed: 05.05.2020).

[17] Dingqi Yang. “Foursquare Dataset”. In: San Francisco, USA, May 2019. URL: https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

[18] Dingqi Yang et al. "Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach". In: *The World Wide Web Conference*. WWW '19. San Francisco, CA, USA: Association for Computing Machinery, 2019, pp. 2147–2157. ISBN: 9781450366748. DOI: `10.1145/3308558.3313635`. URL: `https://doi.org/10.1145/3308558.3313635`.

[19] Guido van Rossum. *Python*. URL: `https://www.python.org`. (accessed: 08.01.2020).

[20] Ajay Thampi. *Reserve Geocode*. URL: `https://pypi.org/project/reverse_geocoder/`. (accessed: 12.04.2020).

[21] *GeoPy*. URL: `https://geopy.readthedocs.io/en/stable/`. (accessed: 12.04.2020).

[22] David Cournapeau. *Sklearn*. URL: `https://scikit-learn.org/stable/`. (accessed: 01.04.2020).

[23] Travis Oliphant. *Numpy*. URL: `https://numpy.org`. (accessed: 01.04.2020).

[24] Eric Jones Travis Oliphant Pearu Peterson. *Scipy*. URL: `https://www.scipy.org`. (accessed: 01.04.2020).

[25] John D. Hunter. *Matplotlib*. URL: `https://matplotlib.org`. (accessed: 05.03.2020).

[26] *Hmmlearn*. URL: `https://hmmlearn.readthedocs.io/en/latest/`. (accessed: 08.04.2020).

[27] Peter Norvig Stuart J. Russell. *Artificial Intelligence A Modern Approach: Third Edition*. 2010, pp. 578–583.

[28] MEJ Newman. "Power laws, Pareto distributions and Zipf's law". In: *Contemporary Physics* 46.5 (2005), pp. 323–351. DOI: `10.1080/00107510500052444`. eprint: `https://doi.org/10.1080/00107510500052444`. URL: `https://doi.org/10.1080/00107510500052444`.

Appendices

Appendix 1 - Maeda et al. parameters variations

T = 2, r = 5, p = 0

Accuracy	Recall	Precision	Average rate	F1 Score
85.24	49.41	36.74	52.26	42.14

T = 2, r = 10, p = 0

Accuracy	Recall	Precision	Average rate	F1 Score
87.03	43.57	40.96	57.19	42.22

T = 2, r = 15, p = 0

Accuracy	Recall	Precision	Average rate	F1 Score
87.66	39.42	42.72	56.6	41.0

T = 3, r = 5, p = 0

Accuracy	Recall	Precision	Average rate	F1 Score
82.43	69.91	34.72	62.35	46.4

T = 3, r = 10, p = 0

Accuracy	Recall	Precision	Average rate	F1 Score
85.15	64.43	38.96	62.85	48.56

T = 3, r = 15, p = 0

Accuracy	Recall	Precision	Average rate	F1 Score
86.51	61.76	41.87	63.38	49.91

T = 4, r = 5, p = 0

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
80.77	79.96	33.78	63.38	47.5

T = 4, r = 10, p = 0

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
83.91	75.55	37.97	65.81	50.54

T = 4, r = 15, p = 0

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
85.45	72.89	40.61	66.32	52.16

T = 2, r = 5, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.06	9.23	48.52	48.94	15.5

T = 2, r = 5, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.15	4.82	51.65	48.53	8.82

T = 2, r = 5, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.18	3.28	54.24	48.9	6.16

T = 2, r = 10, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.21	7.89	52.74	49.95	13.73

T = 2, r = 10, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.21	4.05	55.63	49.63	7.55

T = 2, r = 10, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.18	2.72	55.21	49.04	5.18

T = 2, r = 15, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.24	6.92	54.22	50.13	12.27

T = 2, r = 15, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.22	3.74	57.03	50.0	7.02

T = 2, r = 15, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.17	2.41	55.29	48.96	4.62

T = 3, r = 5, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
88.72	13.74	44.08	48.85	20.95

T = 3, r = 5, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.12	7.48	50.0	48.87	12.95

T = 3, r = 5, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.2	5.48	53.77	49.48	9.95

T = 3, r = 10, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.03	12.56	48.42	50.0	19.95

T = 3, r = 10, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.26	6.77	55.23	50.42	12.06

T = 3, r = 10, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.22	4.66	55.49	49.79	8.6

T = 3, r = 15, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.23	12.2	52.08	51.17	19.77

T = 3, r = 15, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.3	6.25	57.55	51.03	11.28

T = 3, r = 15, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.22	4.41	55.84	49.82	8.21

T = 4, r = 5, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
88.41	16.15	41.56	48.71	23.26

T = 4, r = 5, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.03	8.92	47.67	48.54	15.03

T = 4, r = 5, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.13	6.56	50.39	48.69	11.61

T = 4, r = 10, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
88.77	15.27	45.29	49.78	22.84

T = 4, r = 10, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.14	8.2	50.47	49.27	14.12

T = 4, r = 10, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.23	6.05	54.38	49.89	10.89

T = 4, r = 15, p = 1

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
88.96	14.71	47.67	50.45	22.48

T = 4, r = 15, p = 2

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.23	7.89	53.29	50.14	13.74

T = 4, r = 15, p = 3

Accuracy	**Recall**	**Precision**	**Average rate**	**F1 Score**
89.23	5.69	54.68	49.87	10.31

Table 10. *Maeda et al.* parameters variations

Appendix 2 - Repository link

Repository https://gitlab.cs.ttu.ee/angrun/foursquare-angrun