Instagram-based Recommendation System for Travel Destinations in Italy

Creators
Taize Kang
Fanni Kovapohja
Markku Ylitalo

University of Helsinki
Introduction to Data Science (DATA11001)
18.10.2020

1. Introduction

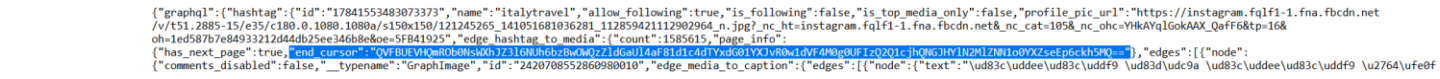
We have created a recommendation system for travel destinations in Italy, which is based on Instagram posts with the hashtag *italytravel.* We scraped 34 578 Instagram posts from the web, collected the locations and hashtags of them and analyzed this information to define the social media popularity and features of 839 different locations. With the analyzed data, we created a recommendation system, in which user can select different features and get location recommendations based on them. The recommendation system is implemented by exploiting k-means clustering and it includes 658 different locations. The final deliverable is a web page consisting of a feature selection window and a recommendation window with the popularity filter and the filtered recommendations on the map of Italy.

2. Web Scraping

We gathered the data from Instagram with the web scraping program written in Python. When creating the web scraper we exploited the ideas of the blog post *Scraping Instagram Locations with Python* written by Ksenia Tikhomirova (https://medium.com/@tiks.dev/scrap-instagram-locations-with-python-d48ba6e56ebc, read on 18.9.2020). The Instagram web page with the hashtag *italytravel* shows approximately 70 Instagram posts at a time in order of publication time where the newest posts are shown first. With the following URL address structure, it is possible to get at the previous loading page consisting of the next newest 70 posts:

https://www.instagram.com/explore/tags/italytravel/?__a=1&max_id=<end_cursor>. (1)

When the parameter <end_cursor> is empty, the URL address (1) leads to the front page of Instagram's subgallery under the hashtag *italytravel.* From that page, it is possible to collect the shortcode of each post for the future processing and also fetch the <end_cursor> value for the next loading page of posts.



(https://www.instagram.com/explore/tags/italytravel/?__a=1&max_id=, taken on 15.10.2020)

When this is done repeatedly, the shortcodes of posts can be collected from multiple loading pages.

The information page of each post can be found by attaching the shortcode of post to the following URL address:

https://www.instagram.com/p/<shortcode>/?__a=1 (2)

The URL address (2) leads to the page where the location information and caption of a post can be gathered. The caption is the part of a post that includes text, user account links and hashtags.

(https://www.instagram.com/p/CGYCnL5jUMx/?__a=1, taken on 15.10.2020.)

On 18.9.2020, we managed to scrape 500 pages of Instagram posts, altogether 34 578 posts. The connection problems of Instagram complicated the scraping so we accepted the largest number of posts we succeeded to get as our data set instead of the planned 150 000 posts. We collected the location name and caption of each post and saved this information to JSON file.

3. Cleaning the Data

Preparing the Data and Cleaning the Locations

We created a new Python module where the JSON file of the raw data was read in. The locations and captions of the Instagram posts were arranged into a Pandas data frame (df1). At first, we lowercased the whole data frame and converted all the data types of the elements to the string format.

```
       image_id                     location                                              text
0             0
1             1       Ponte Vecchio Firenze  Panoramic view of the Ponte Vecchio, from the ...
2             2
3             3        Abbazia di Sant'Antimo  Always so beautiful!\n#abbaye #abbayesantantim...
4             4 Fiordo di Furore - Amalfi Coast  Fiordo di Furore, Amalfi Coast IT \nDouble ta...
...         ...                          ...                                               ...
34573     34573
34574     34574        Burano, Veneto, Italy  #burano #italy #venice #venezia #buranoisland ...
34575     34575             Florence, Italy  Another day, another sunset. This view never g...
34576     34576   San Michele al Tagliamento  Italy IT \n#boattrip #boat #italy #perfectholi...
34577     34577                  Passo Sella  Beautiful Sunset 👌💚\n•\n• \n\n 💚 \n•\n•\n•\n#t...

[34578 rows x 3 columns]
```

```
                           location                                              text
             ponte vecchio firenze  panoramic view of the ponte vecchio, from the ...
              abbazia di santantimo  always so beautiful!\n#abbaye #abbayesantantim...
          fiordo di furore amalfi coast  fiordo di furore, amalfi coast IT \ndouble ta...
                             casoli  casoli, cheiti- italy\n.\n.\n.\n#casoli #chiet...
                             venice  would you believe me if i told you this was th...
                                ...                                               ...
                  la valle della luna  un po' cowboy 👢👒 un po' Ibiza 🍒🌴 \nentrambi i ...
                      burano veneto  #burano #italy #venice #venezia #buranoisland ...
                           florence  another day, another sunset. this view never g...
           san michele al tagliamento  italy IT \n#boattrip #boat #italy #perfectholi...
34577     34577          passo sella  beautiful sunset 👌💚\n•\n• \n\n 💚 \n•\n•\n•\n#t...

[27759 rows x 3 columns]
```

Then we used regular expression methods in the location column to remove all the punctuation marks and to substitute multiple sequent space characters for a single space. After that, we removed all the distinct parts of each location that somehow includes a string 'italy' or 'italia', since all the locations should have been located in Italy based on the *italytravel* hashtag. We also used the 'strip()' method to remove redundant spaces from the beginning and end of the location strings. Finally, we removed all the rows of the data frame (df1) that had not any location value. After the removing, there were 27 759 posts of the previous 34 578 left in the data frame (df1).

We counted the frequencies of each location and examined the 1000 most frequent locations in the data frame (df1). The most frequent locations and the frequencies of them, 'the popularities' from now on, were arranged into a new data frame (df2). The new data frame (df2) had also the empty column for the hashtags of the Instagram posts of each location that would be added later.

```
    loc_id                                   location  popularity  hashtags
0        0                                     venice         937       NaN
1        1                                       rome         674       NaN
2        2                        positano amalfi coast         377       NaN
3        3                                      capri         309       NaN
4        4                                   florence         269       NaN
..     ...                                        ...         ...       ...
953    953                  isola madre isole borromee           4       NaN
973    973  la meridiana relais chateaux on n riviera           4       NaN
940    940                                    sulmona           4       NaN
962    962          la via verde della costa dei trabocchi           4       NaN
936    936                       marina di vasto vasto           4       NaN

[1000 rows x 4 columns]
```

Gathering and Cleaning the Hashtags

We gathered all the captions from the first data frame (df1) that were related to the locations of the second data frame (df2) and added these captions into the empty hashtag column of the second data frame (df2). Each location in the second data frame (df2) had several captions from different posts in the first data frame (df1).

For demonstration, on the right, all the captions of the village of *Monte Sant'Angelo,* which existed in the location column of the data frame (df2), are gathered together. This is an example of the original form of the captions, the information of which was later used to create the features of the travel recommendation tool.

#Puglia #montesantangelo #vieste #pugliagram #puglialove #gargano #italy #italyIT #italytravel #italytrip #centrostorico #the_stag_society #favoboys #hoscos #hoscosmen #hoscosgreece #hmman #italia #instago #landscape #travel #colors #bread #borghi #city #likeforlike #instagram #montesantangelo #italy #italyturism #italytravel #pane #photooftheday #italytrip #ifollow #followers #beautiful #villages #alwaysfollowback #puglia #like4like #architetture #travelingram #italytour #instagood #sicilia #italy #italy_vacations #paesaggiitaliani #italiainunoscatto #visititalia #fotoviaggiatori #lensculturestreets #streetphotography #street #photoforpassion #nikonphotographer #fineartphotography #italy #1415mobilephotographers #streetclassic #landscapephotographi #italytravel #italytrip #italia #italiangirl #italian #paesaggio #paesaggioitaliano #viaggiare #viaggio #tour #tourist #tourism #borghitalia #cittaitaliane #borghitalia #cittaitaliane #fotografandolaterra Instamtanea dal Santuario di San Michele a Monte Sant'Angelo in provincia di Foggia, patrimonio UNESCO dal 2011. #weareinpuglia #inpuglia365 #montesantangelo #foggia #gargano #igersfoggia #igerspuglia #igersitalia #italiait #italiainunoscatto #italia_super_pics #italytrip #italytravel #yallersitalia #ig_italia #yallerspuglia #unplipuglia Buona serata ITITITIT ITALIAITITITIT 👈👈❤❤ puglia_city 11/09/2020⭐⭐⭐⭐⭐⭐⭐⭐ 📷 Autore @pino.di.lecce Luogo #montesantangelo •• ⓒⓞⓝⓖⓡⓐⓣⓤⓛⓐⓣⓘⓞⓝⓢ ···· ·· Amici congratulatevi con l'autore e ammirate la sua splendida galleria• ~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ◈ Thank you so much for Follow: @puglia_city Tag your best photo #puglia_city ~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~ 👓Admin: @massimomorelli33 @luigi70l ..👉 selected by @luigi70l ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ @splash_reflex #splash_reflex @hdr_reflex N O W E B P I C S N O S T O L E N P I C S ❗ Taggate solo foto scattate da voi, vecchie o nuove ma solo originali❗ ⚠Le foto saranno controllate!! È gradito un repost anche solo temporaneo. Grazie! Verranno prese in considerazione solo le foto di chi segue @puglia_city VISITA LE NOSTRE PAGINE AMICHE. ⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐ @hdr_reflex @splash_reflex @bnw_reflexq.⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐ #instago #landscape #travel #sea #nature #instangram #instangramitalia #italia #italy #italyturism #italytravel #follow4follow #photooftheday #like4like #travelingram #instagood #italytour #villages #village #ifollow #beautiful #architetture #all_shots #montesantangelo #montesantangelopuglia #puglialovers #pugliagarganica #gargano #italialovers #italianplaces #borghiitaliani #forestaumbra #foggia #vieste #manfredonia #mareadriatico #adriaticsea #sea #collina #pugliagram #pugliacom #garganocoast #puglia #instapuglia #instaphoto #italy #italytravel #italycoast #pugliatravel Buon pomeriggio ITITITIT ITALIAITITITIT 👈👈❤❤ puglia_city 08/09/2020⭐⭐⭐⭐⭐⭐⭐⭐ 📷 Autore @micheletodisco_ph Luogo #montesantangelo ••• ⓒⓞⓝⓖⓡⓐⓣⓤⓛⓐⓣⓘⓞⓝⓢ ···················· ·································· Amici congratulatevi con l'autore e ammirate la sua splendida galleria• ~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~ ◈ Thank you so much for Follow: @puglia_city Tag your best photo #puglia_city ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ 👓Admin: @massimomorelli33 @luigi70l ..👉 selected by @luigi70l ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ @splash_reflex #splash_reflex @hdr_reflex N O W E B P I C S N O S T O L E N P I C S ❗ Taggate solo foto scattate da voi, vecchie o nuove ma solo originali❗ ⚠Le foto saranno controllate!! È gradito un repost anche solo temporaneo. Grazie! Verranno prese in considerazione solo le foto di chi segue @puglia_city VISITA LE NOSTRE PAGINE AMICHE. ⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐ @hdr_reflex @splash_reflex @bnw_reflexq. ⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐⭐ #instago #landscape #travel #sea #nature #instangram #instangramitalia #italia #italy #italyturism #italytravel #follow4follow #photooftheday #like4like #travelingram #instagood #italytour #villages #village #ifollow #beautiful #architetture #all_shots La storia è testimonianza del passato, luce di verità, vita della memoria, maestra di vita. 🏰 #castello#castle #castlephotography #templari#montesantangelo#loveit#loveitaly#nonveniteinpuglia#storia#storiaitaliana#paradisopugliese#terrasanta#italytravel#unicaalmondo#solocosebelle#postiincantevoli#peaceful#istagramphoto

We used regular expression methods to remove all the other parts of the caption strings but the hashtags. We only saved the strings starting with '#' and containing at least 3 alphabetical or '_' characters. Then we removed all the '#' marks and created a list of distinct hashtag words for each post with the 'split()' method. We decided to use only hashtags and not the other parts of the captions, because the hashtags are more homogeneous than the other parts, which include more different languages and wordings, and typographical errors.

After that, with the help of programmed automatization and the self-collected 'stop words' of hashtags, for example, 'blue', 'usa', 'lol', 'sky', 'follow', 'follow4follow', 'woman', 'friday', 'morning', 'couple', 'brunette', 'adventure', 'september', 'settembre' and 'selfie', we managed to remove the most of the uninformative hashtags in the data frame (df2). We also removed all the hashtags that were as a part of the string of any location mentioned in the data frame (df2), because the location information in hashtags does not have any additional value when we already know the locations. For the comparison between the previous and current state, there is the cleaned version of the hashtags of *Monte Sant'Angelo* shown under:

```
['vieste', 'gargano', 'centrostorico', 'the_stag_society', 'hoscos', 'hoscosmen', 'hoscosgreece', 'hmman', 'landscape', 'bread', 'borghi', 'city', 'pane', 'villages', 'architetture', 'paesaggio', 'tour
', 'foggia', 'gargano', 'splash_reflex', 'landscape', 'sea', 'nature', 'villages', 'village', 'architetture', 'gargano', 'forestaumbra', 'foggia', 'vieste', 'manfredonia', 'mareadriatico', 'adriaticsea
', 'sea', 'collina', 'garganocoast', 'splash_reflex', 'landscape', 'sea', 'nature', 'villages', 'village', 'architetture', 'castello', 'castle', 'templari', 'storia', 'paradisopugliese', 'terrasanta',
'unicaalmondo', 'solocosebelle', 'peaceful', 'splash_reflex', 'landscape', 'sea', 'nature', 'villages', 'village', 'architetture']
```

Duplicates of the Locations

We noticed that in the data frame (df2), there were locations that meant the same but had been written differently or in different languages, for example, in English, Italian, German, Spanish or Chinese. We tried to exploit programmed automatization when searching the duplicates as much as we could, but still, there was a lot of manual work to do in this process. When we combined these duplicates, we also combined all the hashtags and summed all the popularity values of them. After this process, there were 892 locations left.

	location	popularity
0	venice	1230
1	rome	867
2	positano amalfi coast	543
3	capri	350
4	amalfi coast	322
..	...	...
887	hotel puccini	4
888	giovinazzo bari puglia	4
889	villa tellus holiday villa	4
890	rosignano solvay toscana	4
891	sulmona	4

[892 rows x 2 columns]

Later we realized that there still were many duplicates left. There were also unwanted locations outside the boarders of Italy. We examined the locations again more thoroughly than the first time. We dealt with the duplicates in the same manner as above and removed the unwanted locations from the data frame (df2). Also, the rows without any hashtags were deleted at this point. We created a new cleaned data frame (df3), where there were 839 locations left.

4. Features of the Locations

We gathered all the hashtags (*) from the data frame (df3) and counted the 200 most common hashtags of them. While analyzing the list of the most common hashtags, we decided to divide the hashtags into 7 different classes and 42 subclasses. For example, the class *NATURE* consists of subclasses *beach, lake, river, sea, waterfall, hills ,mountain* and *volcanic,* and the class *ACTIVITIES* consists of subclasses *hiking, sailing, swimming, fashion, luxury, food, restaurant* and *wine.*

For each subclass, we formed a sublist of hashtags from the list of all the hashtags (*) with the help of **fastText**. Each sublist included only hashtags that were related to the given class. For example, fastText suggested that the following hashtags describe the subclass *hiking* the best*:*

```
[('hikingtrails', 0.9842933416366577), ('hike', 0.9762042760848999), ('dolomitemountains', 0.9707328081130981), ('wild', 0.970599889755249), ('dolomitesmountain', 0.964785099029541), ('mountainscapes',
 0.9641682505607605), ('dolomite', 0.9636821746826172), ('dolomitesunesco', 0.9633682370185852), ('mountainstones', 0.9630481004714966), ('dolomitibellunesi', 0.9619472026824951), ('dolomitici', 0.9616
671800613403), ('mountainscenery', 0.9614038467407227), ('dolomiten', 0.9602888822555542), ('dolomitesyou', 0.9602440595626831), ('dolomites', 0.9588807821273804), ('dolomitiamo', 0.9587298631668091),
('dolomitiunesco', 0.9568751454353333), ('dolomitas', 0.9561613202095032), ('mountainsarecalling', 0.9558297395706177), ('wilderness', 0.9553650617599487), ('dolomiti', 0.9549998641014099), ('cinquetor
ricortinadampezzo', 0.9541064500808716), ('mountains', 0.9533324837684631), ('cortinadampezzo', 0.953248143196106), ('dolomitidasogno', 0.95317143201828), ('secedamountain', 0.9528548121452332), ('moun
tainscape', 0.9520748853683472), ('trekking', 0.9518843889233645), ('natgeowild', 0.9514551758766174), ('dolomitisuperski', 0.9500757455825806), ('mountainbike', 0.9496070146560669), ('folkscenery', 0.9
475473165512085), ('addictedtothemountains', 0.9470806121826172), ('secedam', 0.9468228816986084), ('trecimedilavaredo', 0.9458196759223938), ('dolomity', 0.945677638053894), ('letrecimedilavaredo', 0.
945537805557251), ('dolomitiemotions', 0.9453480243682861), ('wildernessnation', 0.9446677565574646), ('dolomitipatrimoniounesco', 0.9440531730651855), ('mountainview', 0.9422052502632141), ('visitdolo
mites', 0.9415383338928223), ('mountainside', 0.9413654208183289), ('mountaineering', 0.9398941993713379), ('mountaintop', 0.9398379921913147), ('cinquetorri', 0.9394592046737671), ('mountainviews', 0.
9391295909881592), ('dreizinnen', 0.9378666877746582), ('hiker', 0.9376326799392 7), ('mountainelopement', 0.9375296831130981)]
```

We added 42 empty columns to the data frame (df3) for each subclass. We created a program that went all the locations and features through one by one, so that the frequency of each feature in each location's

hashtag list was counted. For example, for the location *Venice* and subclass *sailing,* the program increased the value at the row *Venice*, in the column *sailing,* every time when some hashtag from the fastText processed hashtag list of *sailing* was found from the hashtag list of *Venice.* At this point, the hashtag lists of locations could have contained the same hashtag multiple times, so that the frequencies of features would be counted correctly when considering that the hashtag list of each location had been formed from several Instagram posts.

The frequencies were altered to relative frequencies along the row/location, so that the locations with different popularities would be comparable with each other. The most popular locations had the largest absolute frequencies of all of the features, so the less popular locations would never be recommended without relative frequencies. After that, the column of hashtag lists of locations was removed, and we had a final data frame (df4) for building the recommendation system:

	loc_id	location	popularity	hiddengem	unesco	renaissance	gothic	ancient	medieval	...	fashion	aesthetic	luxury	food	art	history	culture	archeology	architecture
0	0	venice	1230	0.000	1.515	1.732	0.649	0.000	0.000	...	2.386	0.386	2.492	1.030	7.595	3.331	6.162	4.420	7.934
1	1	rome	867	0.000	0.235	0.469	0.000	3.521	0.000	...	1.030	0.263	1.459	1.214	8.077	5.110	8.243	5.101	9.032
2	2	positano amalfi coast	543	0.000	1.064	0.000	0.000	1.064	0.532	...	1.942	0.161	2.604	1.058	2.641	1.528	2.048	1.491	2.678
3	3	capri	350	0.000	0.329	0.000	0.000	0.000	0.329	...	1.838	0.060	6.956	0.936	1.821	0.558	1.142	0.739	1.340
4	4	amalfi coast	322	0.704	2.113	0.000	0.000	0.000	0.000	...	2.252	0.081	2.948	1.265	1.497	0.847	1.625	0.754	1.776
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
834	834	marina di vasto vasto	4	0.000	0.000	0.000	0.000	0.000	0.000	...	2.273	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
835	835	loro ciuffenna	4	0.000	0.000	0.000	0.000	0.000	0.000	...	0.962	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
836	836	mantua	4	0.000	0.000	0.000	0.000	0.000	50.000	...	0.000	0.000	0.000	0.000	11.538	7.692	9.615	7.692	11.538
837	837	abbazia di santantimo	4	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	0.000	0.000	44.444	16.667	0.000	5.556
838	838	sulmona	4	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	39.286	0.000	0.000	7.143	0.000	7.143

[839 rows x 45 columns]

5. K-means Clustering for the Locations

The location data was not classified, so we tried to find a solution to divide the location data into classes based on different features. We planned to build a recommendation system, in which user can choose desired features and get location recommendations, with the help of these classes. We could have got better recommendation accuracy if we had exploited the exact relative frequencies of the features. However, within the time limit, we did not discover any clustering method, in which we could have used the relative frequencies. That is why, after experiments, we changed the feature data into binary form by setting values less than 2% to be 0 and the values greater than or equal to 2% to be 1. We interpreted that the relative frequencies less than 2% were not significant.

We decided to use k-means clustering method of Scikit-learn to get k classes of locations based on their feature similarity. After experimenting different k-values, we decided to set k = 15, because the sizes of the clusters were the most appropriate then. If we have more locations in the future, the value of k must be determined again. The first step was to set 15 centroids randomly and divide the data into 15 classes by them. Then the data was re-classified by calculating the distance until there was no change to the centroids. After training the data, we found that there were two very large clusters, which would make the recommendations inaccurate. We used k-means again for them with k = 4 to get smaller clusters. Unfortunately, this worked only with the one large cluster. The other large cluster was divided into one very large subcluster and three small subclusters, sizes of 2, 1 and 1. That is why only another one of the large clusters was divided into smaller clusters, which may affect negatively the recommendation results.

6. Recommendation System

The purpose of a recommendation system for travel destinations is to find locations similar to the preferences of a user from the data. We implemented this as follows: User decides whether or not he wants to select each feature. If he selects the feature, the value of this feature is 1, otherwise the value of the feature is 0. After getting the input list from user, we use k-means to examine in which class the list should belong. If the class is the large class, which is divided into smaller classes, we use k-means model again to get smaller and more accurate classification. When the right class is found, the whole class is recommended for the user.

7. Final Deliverable

We decided to use a web page as a front end and our recommendation system as a back end. We exploited Flask to do that. The web page consists of two windows. In the first window, user can select desired features by ticking the checkboxes. Then we transfer the checkboxes to a list of zeros and ones, which we use as an input of the k-means model. When we got the result class, the locations of the class are shown on the map of Italy, which is the second window. The map of the second window is implemented with Plotly. We exploited GeoEarth's API to get the coordinates of the locations. However, some of the coordinates were not found, so we had to remove these locations from the recommendation system. In the final deliverable, there are 658 different locations left. The dots of different sizes on the map refer to the popularity of locations — the bigger dot means more popular location. There is also a filter beside the map for user to select four different levels of popularity. These levels are shown on the map with different colors of the dots.

8. Evaluation and Conclusions

In the project proposal, we discussed that we were going to scrape Instagram posts with the hashtag *italytravel* from the web, gather the locations of the posts, use the number of times a location appears in the posts as the popularity of it and sort the locations using hashtags. We also planned to build a recommendation system for travel destinations, in which user can filter the popularity and type of the destinations. All that is what we ended up to implement. Also the visualizations were carried out as planned, apart from showing the list of the recommended locations. We thought that it is clearer for the end-user to see the locations with their names only on the map. What we did not implement from the project proposal, was exploiting weather or average cost information of the locations. We decided to ignore these features, because the differences in them between different locations are not significant inside Italy. Because we were only able to collect the 34 578 latest Instagram posts, the most of them were published in September. That is why we could not exploit the season or dates of the posts as we planned. In the future work, it would be better to analyze more posts and distribute the gathering of posts around the year to make the data sample more diverse. In addition, we did not use the decision forest model we wrote about because we had only unlabeled data. The k-means clustering turned out to work better for our project.

Since the tool was designed to offer user optimized travel recommendations finding the both viral and hidden gems of Italy, it fullfills its purpose laudably. There are already many travel guides available, but because the most popular destinations gather all the hype, the lesser-known ones are often almost impossible to find. The added value of the tool is that users will find new and interesting destinations that they may not have found without it. That makes our idea to exploit Instagram posts unique and useful. The visual outcome of the system is functional and helps user to locate the destinations easily. There are also many expanding options, for example, a road trip organizer, which connects desired locations. In addition, there are a couple of things to improve: k-means do not perform accurately with all combinations of the features, there are still a few locations outside Italy included in, and program needs to be ran again for another search. Also the gathered data may not be so reliable due to the impacts of the COVID-19 situation on traveling in Italy.

Taize Kang majored his BSc degree in intelligent science and technology, and Fanni Kovapohja and Markku Ylitalo majored their BSc degrees in mathematics. Because the project hardly included any mathematics, and Kovapohja and Ylitalo had computer science only as a minor subject in their BSc degrees, nearly everything of the project, except from the basic knowledge of the Python programming, was new to them. Kang had more experience in programming but was not very familiar with the front end process and there were new libraries for him. None of us was very familiar with the machine learning techniques. Creating the idea of the project and making the whole recommendation system from scratch required huge efforts from the whole group. That is why the learning outcomes of this project and course were really massive. Each group member had equal workload and participation in every stage of the process. We had a lot of work to do in all stages of the data life cycle, so our project is suitable for this course and includes sufficient amount of work.