VIRTUAL TRIP AND PLANNING ADVISOR

Applied Data Science Capstone Project
IBM Data Sceince Professional Certificate Specialization
By Ihor Kulmatytskyy

Virtual Trip advisor or classifier project consists of:

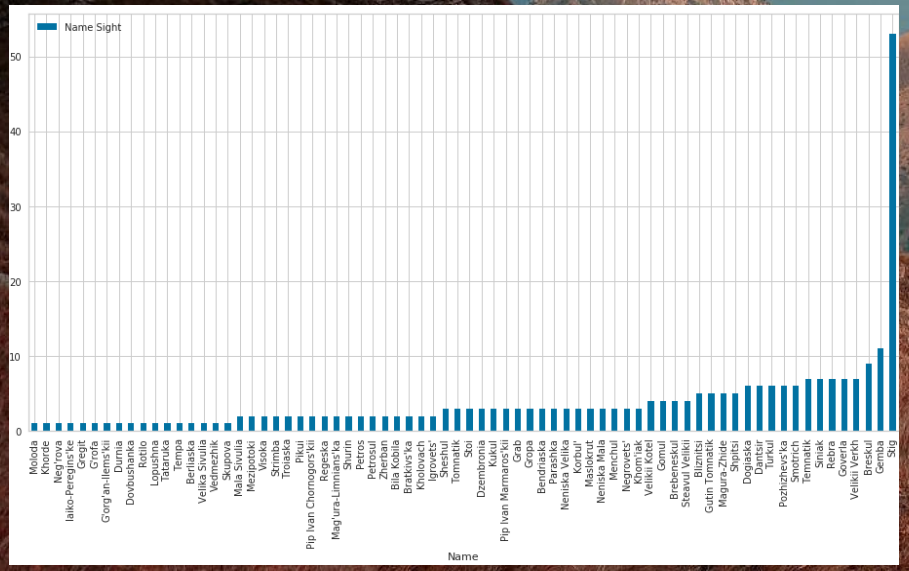
- Gathering data about the Ukrainian Carpathian Mountains and nearest venues
- Applying clustering with k-Means to determine mountain peaks with similar nearest outdoor activities
- Studying result, picking labels to the determined clusters, visualization, and presentation

	Name	Height	Location	id	Latitude	Longitude
0	Goverla	2061.0	Chornogora	1	48.15111	24.50000
1	Brebeneskul	2035.8	Chornogora	2	48.09833	24.58056
2	Pip Ivan Chornogors'kii	2028.5	Chornogora	3	48.04778	24.62778
3	Petros	2022.5	Chornogora	4	48.17194	24.42111
4	Gutin Tomnatik	2016.4	Chornogora	5	48.10000	24.55667

Carpathian mountain peaks from Wikipedia

Sights and venues from Foursquare

	id	Name	Category	Latitude	Longitude
0	1	Говерла (2061 м) / Hoverla	Mountain	48.16039	24.50037
1	2	Брескул (1911 м) / Breskul	Mountain	48.15037	24.51103
2	3	Гірське Крісло	Other Great Outdoors	48.16272	24.50573
3	4	Говерлянка	Mountain	48.16615	24.50546
4	5	Пожижевська (1822 м)	Mountain	48.14430	24.52357

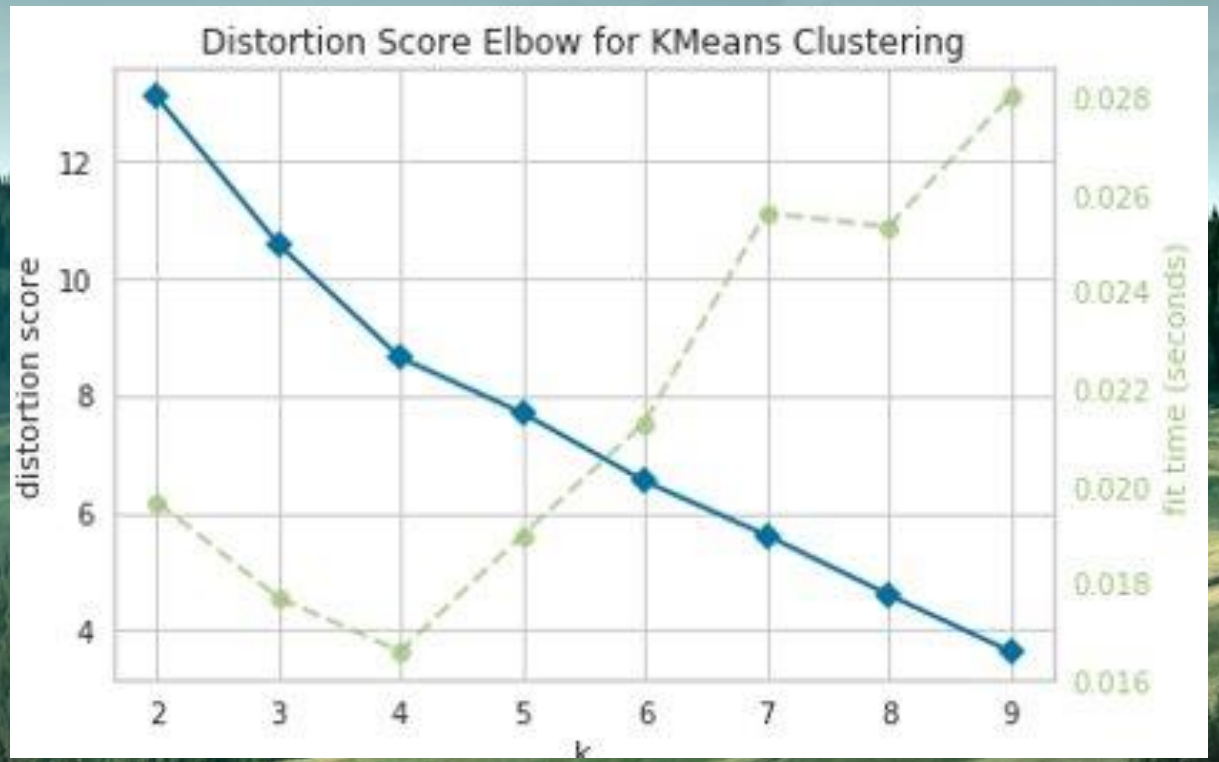


Frequencies of venues in relation to mountain peaks – illustration of non-consistent data on small data amounts

Gather data from and prepare it to further processing

Apply k-Means to cluster Mountain Peaks by categories of the nearest venues.

1. Use the elbow method to determine an optimal number of clusters k = 6.
2. Perform k-Means and pick up proper cluster names
3. Display result on a map



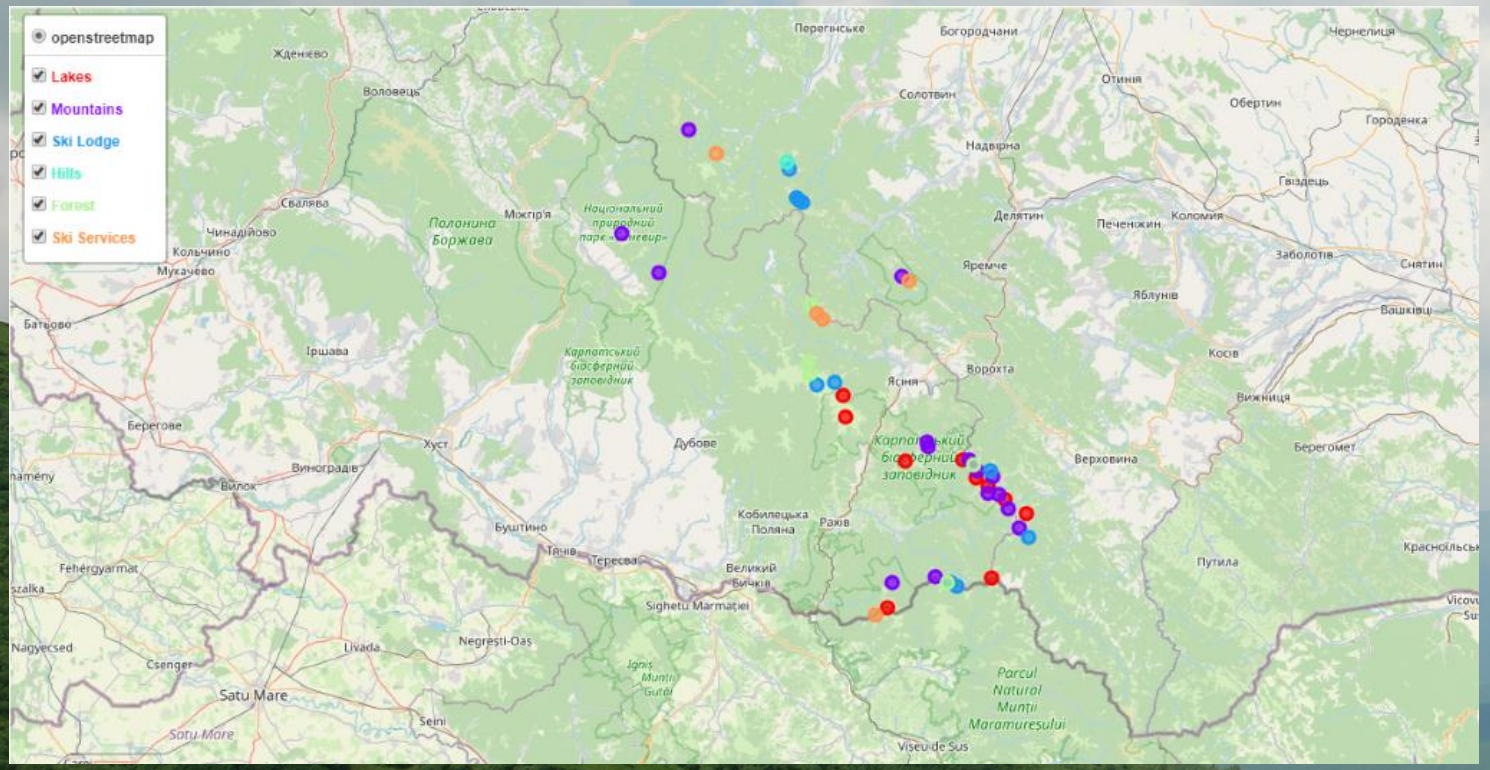
Use the elbow method to determine an optimal number of clusters k = 6.

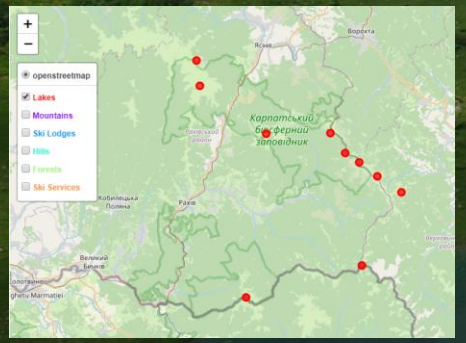
Perform k-Means and pick up proper cluster names

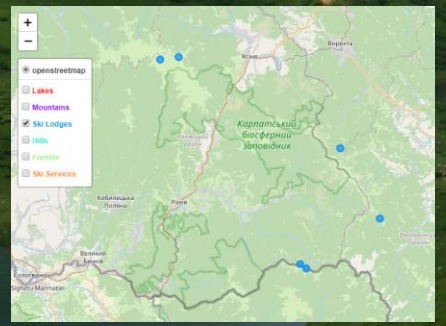
In [55]:
```
# set number of clusters
k = 6

# run k-means clustering
kmeans = KMeans(n_clusters=k, random_state=0).fit(sights_categories.drop(['Name'], 1).drop

# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

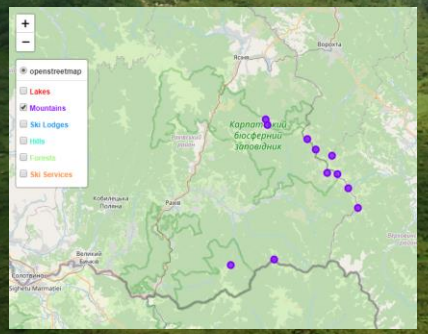
Out[55]:
```
array([0, 5, 1, 1, 1, 2, 5, 4, 1, 5, 2, 0, 5, 1, 2, 2, 2, 0, 1, 1, 1, 4,
       2, 1, 1, 1, 0, 4, 0, 0, 1, 2, 0, 2, 0, 1, 4, 4, 0, 1, 2, 2, 3, 5],
      dtype=int32)
```

In [67]:
```
cluster_names = ['Lakes', 'Mountains', 'Ski Lodges', 'Hills', 'Forests', 'Ski Services']
```









Display result on a map with an option to filter by cluster

THANK YOU

Stay healthy and positive :)