Road - Map

The search for a representative route problem is reduced to two strategies with different solutions:

the first strategy is based on finding the most popular route and uses algorithms to find the maximum flow, but the problem consists in correctly estimating the weight of each edge of the graph and it requires an evaluation of at least O (VE log(V2/E)) with the use of dynamic programming and one of the disadvantages is the determinism of the problem.

the second strategy uses the KNN algorithm search for nearest neighbors and partition into clusters this algorithm is not determined and gives a result, the only difficulty is the choice of the number of clusters here we proceed from an empirical estimate

Based on the second chosen strategy, we can observe that there is no more than one route in different directions, but if the routes are correctly divided into clusters, they should be visually similar, and the distance between points should not exceed the error. For outliers we will use the interquartile range.

Python 3.8, the KNN algorithm, and seaborn packages for visualization, pandas for dataframe and numpy are used to solve the problem.   
  
The choice of language and method is due to the simplicity of implementation and the amount of time spent.

What would be nice to improve, use a more modern algorithm, give more accurate static characteristics and analysis, but this requires the power of for example AWS, AZURE, etc.   
We need to add unit tests for our model.

We should also think about delivering to the client through containerization and dependency management. Why Java was not selected, unfortunately the language does not have a good API for working with data with the exception of Spark technology which has MLlib, but unfortunately there is not enough qualification in using this library.