Team:

1. Mahbubur Rahman (914626),
   M.Sc in Data Science
   Berliner Hochschule for Technik, Berlin, Germany.

Use Case:
Help the City Council to take proper decision on designing parking spots.

Questions:
To solve the above use case, I will answer the following questions using two different machine learning methods and then compare the results of those two machine learning methods.

1. Which explanatory variables have impact or influence on parking spots estimation?

2. How many parking spots are needed for a specific lor or area or parking lane? (Calculation using the explanatory variables from question 1)

Dataset:
Geospatial datasets of districts Friedrichshain-Kreuzberg and Neukölln.

Dataset Description:
There are many datasets from the district office. So I explored and analysed all datasets and after that, I selected the following datasets to answer above questions.

1. **estimated_parking_spots_kfz.geojson**

Variable Name	Data Type
highway	string
name	string
parking	string
orientation	string
position	string
capacity	integer
source_orientation	string
source_position	string
source_capacity	string
geometry	geometry object

2. **registered_cars.geojson**

Variable Name	Data Type
district	string
lor	string
Bezirksregion	string
Prognoseraum	string
lor size in m²	float64
inhabitants_total	int64
of_those_inhabitants_18+	int64
vehicles_overall	int64
cars_only	int64
vehicles_per_1000_inhabitants	int64
cars_per_1000_inhabitants	int64
geometry	geometry object

3. **kfz_lor_planungsraum.geojson**

Variable Name	Data Type
Schlüssel	integer
Bezirk	string
Planungsraum	string
Bezirksregion	string
Prognoseraum	string
Flächengröße in m²	float64
Einwohner insgesamt	int64
darunter 18 Jahre und älter	int64
Kraftfahrzeuge insgesamt	int64
darunter Pkw	int64
Kfz pro 1000 EW	int64
Pkw pro 1000 EW	int64
geometry	geometry object

4. **parking_way.geojson**

Variable Name	Data Type
condition	string
parking	string
orientation	string

condition:other	string
position	string
condition:other:time	date time
source:capacity	string
oneway_direction	string
highway	string
highway:name	string
maxstay	string
capacity	int64
osm-location	string
osm_id	string
id	int64
length	float64
geometry	geometry object

I joined them using their geometry and create 2 datasets from them. I need to do more data preparation on 2 spatially joined datasets. I attached the code file as pdf with this email. This file contains an overview of the datasets.

Machine Learning Methods:

1. Tree Models or Ensemble Methods
2. Support Vector Regression or Neural Networks

I will work on the following language, tools and libraries:

a. Jupyter Notebook,
b. Python,
c. GeoPandas
d. Sklearn

Moreover, according to your suggestion, I have plan to use standard autoML package and AutoGluon. After that, I will give you a review of those packages.

Work Plan:

I follow the CRISP-DM process for this project. So I am doing the following steps:

1. Business Understanding,
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

Completed:

I completed step 1 to 3. The output of those steps are given below:

Business Understanding:

This are written in the “Use Case” and “Questions” sections in above.

Data Understanding and Preparation:

From the above mentioned datasets, I see the patterns and other similarities then I created a single dataset from those datasets. The newly created dataset contains various important information of the LOR(Lebensweltlich Orientierte Räume), vehicles, inhabitants and parking spots of the Friedrichshain-Kreuzberg and Neukölln districts. These information will help to create a solution for the above mentioned use case and also help to answer the above mentioned questions.

Dataset Description:

Variable Name	Data Type
highway	string
name	string
parking	string
orientation	string
position	string
capacity	integer
source_orientation	string
source_position	string
source_capacity	string
geometry	geometry object
index_right	int64
district	string
lor	string
Bezirksregion	string
Prognoseraum	string
lor size in m^2	float64
inhabitants_total	int64
of_those_inhabitants_18+	int64
vehicles_overall	int64
cars_only	int64
vehicles_per_1000_inhabitants	int64
cars_per_1000_inhabitants	int64
geometry	geometry object

ToDo:

1. Create baseline for the above mentioned use case
2. Apply machine learning methods which is described in the “Machine Learning Methods” section.
3. See the results of the two different machine learning methods.
4. Compare the results of those two machine learning methods.