1 Introduction

The database contains information on traffic accidents from 2005. to 2016. in France. For each traffic accident, detailed information was recorded-conditions of the accident and the factors that influenced the accident.

We first get to know data through a series of graphs and tables. Then we look for and process the missing and zero values. We create a universal table on which we apply clustering algorithms, analyze the obtained clusters and draw certain conclusions.

2 Exploring data

Source data can be found here. The file contains 5 tables in *.csv* format. First table (**caracteristics.csv**) has 840,000 rows with 16 attributes, where each row represents one traffic accident. Attributes mainly relate to the time, place and general conditions of the accident. Second table (**holidays.csv**) contains the dates and names of all major holidays in the period between 2005 and 2016. Tabel **places.csv** contains more detailed information about the scene of the accident and the characteristics of the road. Tabels **users.csv** i **vehicles.csv** contain data on participants in traffic accidents and types of vehicles that participated in accidents. The table with participants has 1.88m rows with 12 attributes. Attributes are mostly categorical, where numbers encode different options. An example is an attribute **lum**, from table caracteristics.csv, which contains data on road lighting at the time of the accident.

- 1 - Day
- 2 - Twilight or dawn
- 3 - Night without public lighting
- 4 - Night with public lighting not lit
- 5 - Night with public lighting on

Part of the data can be seen in the picture 1.

	AccidentID	year	month	day	hour	lighting	city/countryside	intersection	atmosphericCondition	collisionTyp
0	201600000001	16	2	1	1445	1	2	1	8.0	3.0
1	201600000002	16	3	16	1800	1	2	6	1.0	6.0
2	201600000003	16	7	13	1900	1	1	1	1.0	6.0
3	201600000004	16	8	15	1930	2	2	1	7.0	3.0
4	201600000005	16	12	23	1100	1	2	3	1.0	3.0

Figure 1: Data

2.1 Traffic accidents analysis

We first look at the number of traffic accidents over the years and number of deaths.

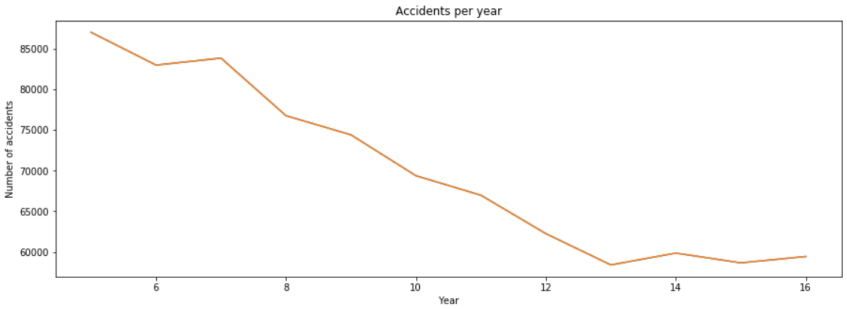


Figure 2: Number of accidents per year

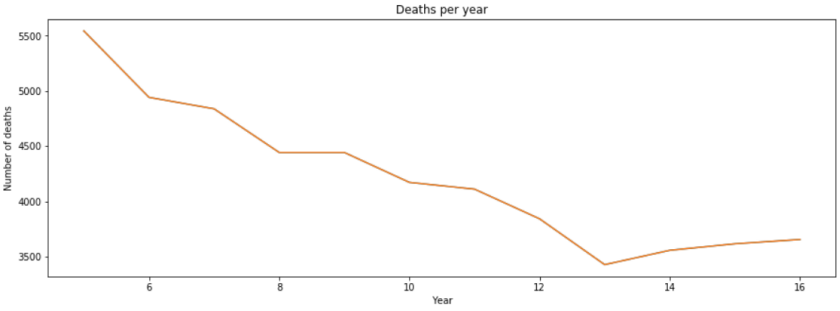


Figure 3: Number of deaths per year

From the graphs we can see that the number of traffic accidents is closely related to the number of fatal outcomes. The number of accidents during the years generally decreases, until 2013. when it reaches a minimum. After 2013, the number of accidents is slightly increasing as is the number of deaths.

By observing the time periods, when traffic accidents occurred, certain conclusions can be reached. From the graphs 4 and 5 we see that a larger number of traffic accidents happen during the summer months (with the exception of August) and in autumn than in winter and spring. When we look at per day of the week distribution, there are fewer accidents on Mondays, and a higher number of accidents on Fridays compared to the average working day. The number of accidents decreases on weekends, particularly on Sundays.

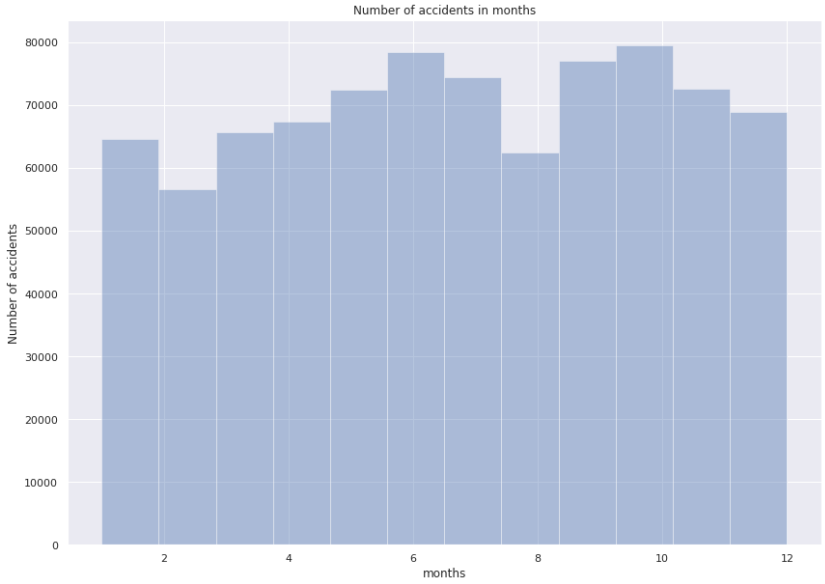


Figure 4: Number of accidents per month

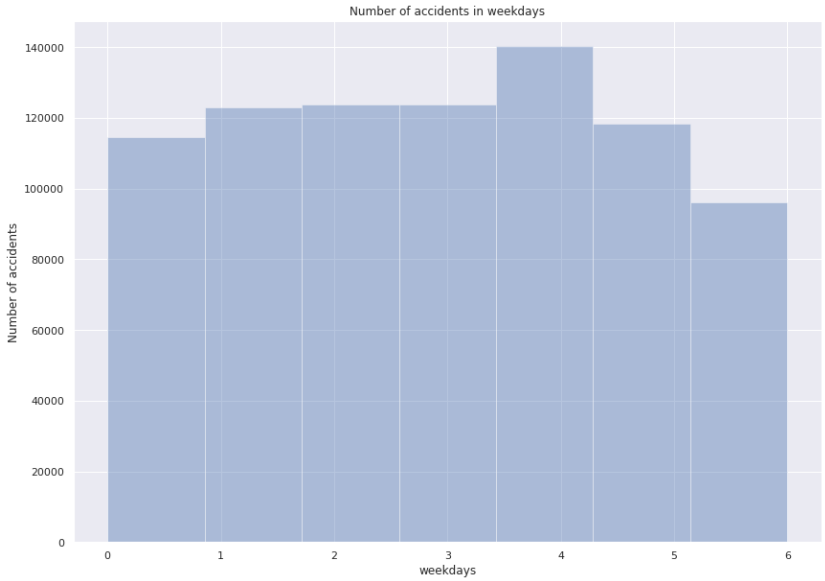


Figure 5: Number of accidents per weekday

Since a similar number of accidents occur on workdays, we will create a new attribute that indicates whether it is a workday or not.
We also check the distribution of traffic accidents by districts. In Figure 6 we see that most accidents occur in a couple of the largest cities in France.

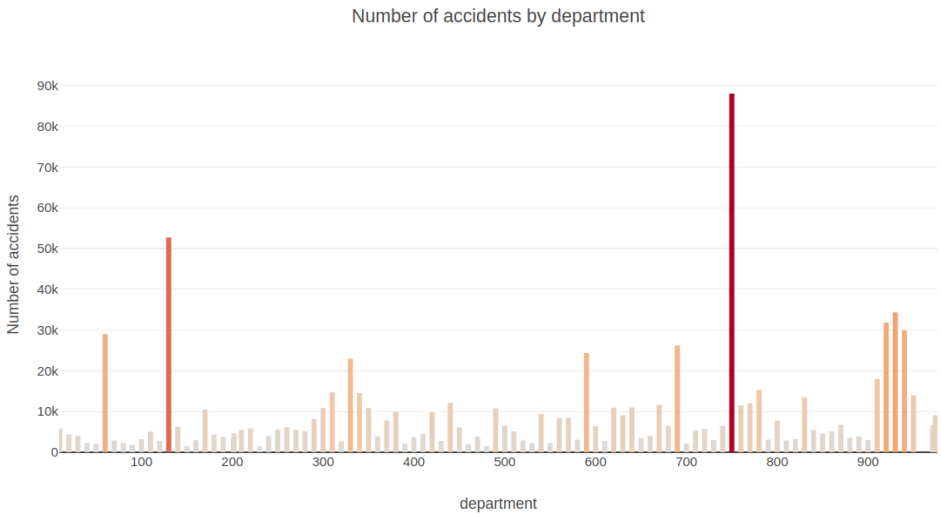


Figure 6: Number of accidents by district

2.2 Missing values

Before merging desired rows into the one file, we will check whether there are missing values, or elements outside the boundaries. First we look at the table **caracteristics.csv**.

Table 1: Missing values

Attributes	**Percentage**
longitude	56.84%
latitude	56.84%
gps coding	56.4%
address	16.73%
collision type	0.0013%
municipality	0%
AccidentID	0%
year,month,day	0%
hour	0%
lighting	0%
city or countryside	0%
intersection	0%
department	0%

The attributes longitude, latitude and gps-coding have the most missing values. Since the above 3 attributes have more than 35% of missing values, they will not be used in further research. Idea is to determine some general conditions in which serious accidents occur (accidents that result in serious injuries or death), attributes related to specific locations are not of interest (in

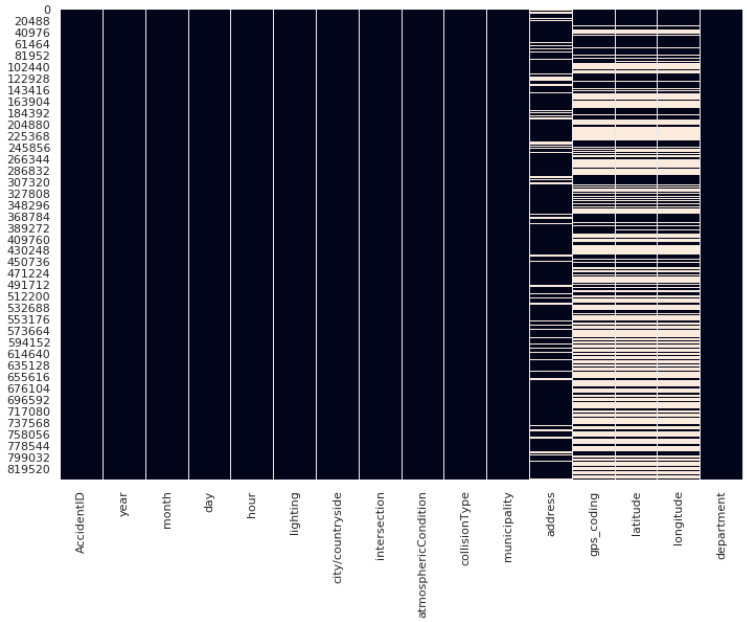


Figure 7: Missing values in table caracteristics.csv

the figure 6 we see that the largest number of traffic accidents happen in large cities). For this reason, we also omit the address, municipality and department attributes.

One of the problems are zero values. In categorical attributes, the different options are coded in range from 1 to n, so zero has no meaning attached to it. For that reason, it is necessary to see how many attribute have zero values and whether they make sense for specific attribute.

Table 2: Zero values

Attributes	**Number of zero values**
latitude	117839
intersection	106
other	0

The latitude attribute has a large number of zero values, because the Greenwich meridian passes through France, while the intersection attribute has no meaning for the value 0, and we must consider this values as missing.
Similarly for tables **holidays.csv**, **places.csv** and **users.csv**.

Table 3: Missing values in **holidays.csv**

Attributse	Percentage
ds	0%
holiday	0%

The **holidays.csv** table, as expected, has no missing values. The table shows the names and dates of all holidays in the observed time period in France.

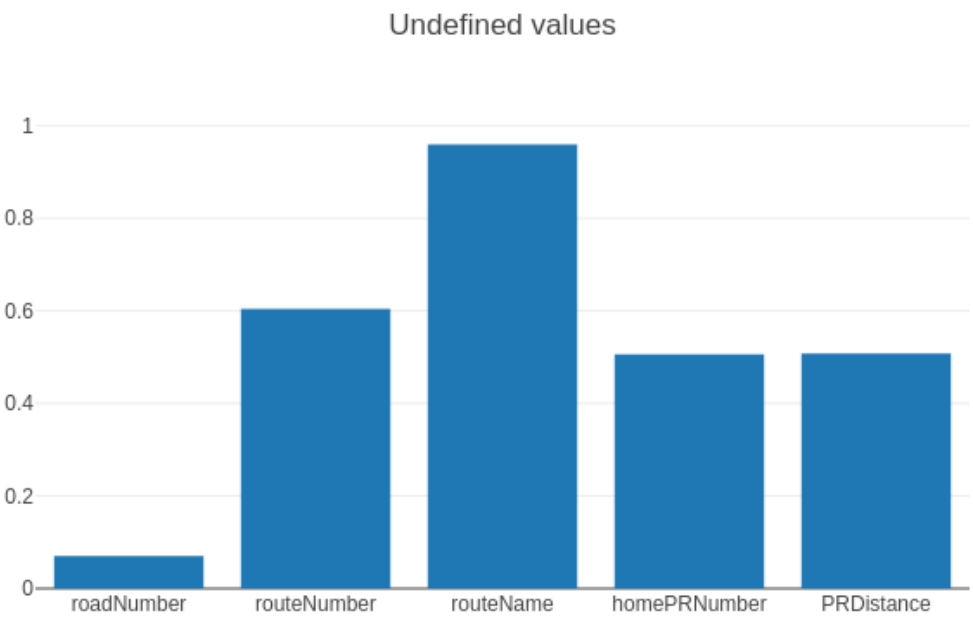


Figure 8: Missing values in places.csv

From the image 8 we see that in the table **places.csv**, the values of the number and name of the road are missing the most. This is not a problem as our intention is to pay more attention to the more general conditions than to the specific places, districts and roads where the accidents occurred.

Unlike the table **caracteristics.csv**, table **places.csv** has a large number of zero values (as we can see 9), which we have to process in some way.

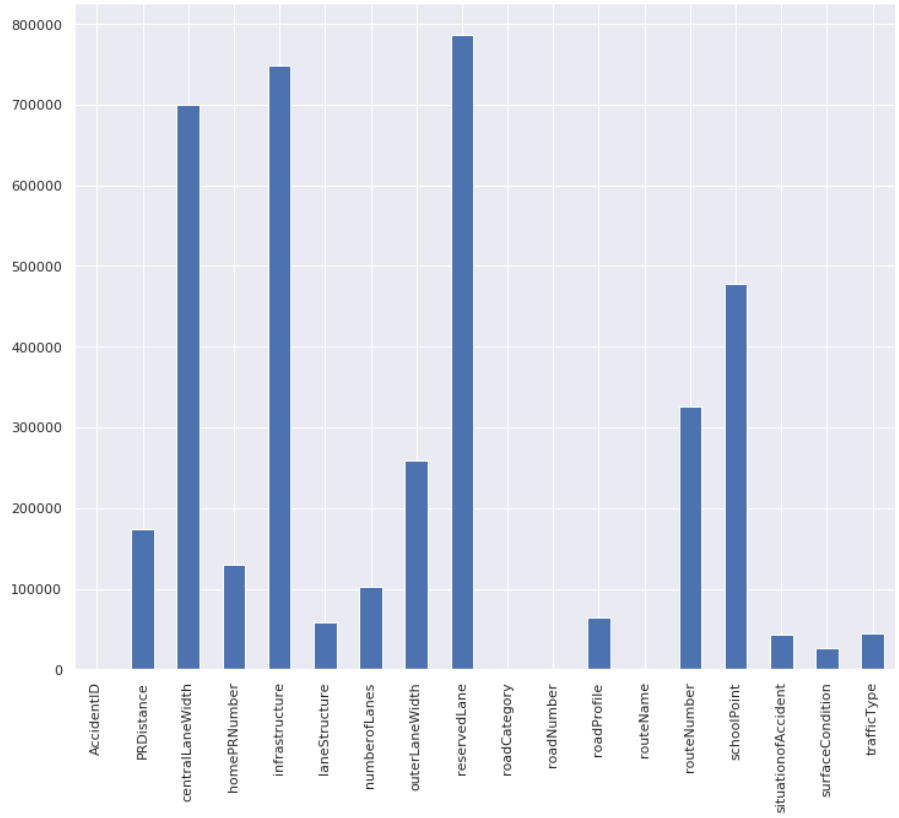


Figure 9: Zero values in places.csv

Finally, the missing and zero values of the **users.csv** table. A large number of zero values is in the attributes that explain the actions and positions of pedestrians, which indicates the absence of information or the impossibility of its presentation.

Table 4: Missing values **users.csv**

Attributes	**Percentage**
place	5.35
safetyEquipment	2.3165
yearOfBirth	0.12
other	<0.1

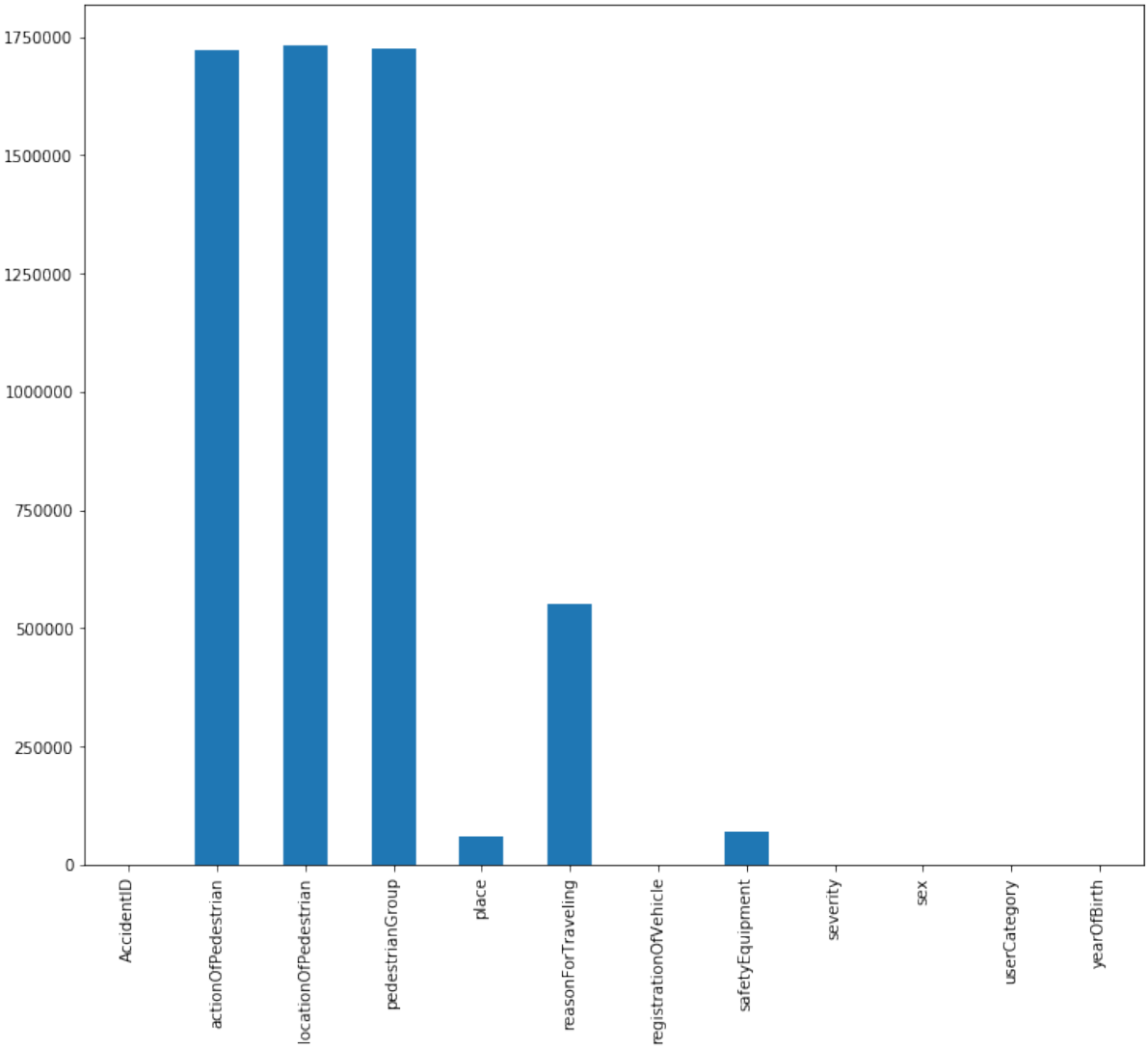


Figure 10: Zero values users.csv

3 Preprocessing

We load the tables in .csv format using the Pandas library, and chang the attribute names (which are mostly abbreviations in French) to make them more understandable.
In the figure 4 we see that in the winter months there are fewer traffic accidents than in the summer, with the exception of August, which is at the level of February (the least days, and therefore the least accidents). This phenomenon occurs because traditionally vacation days are used in August, when French also have collective vacations, which is why there is a sharp drop in accidents this month.

Due to the similarity of the number of accidents within the seasons, we combine the months, and make an attribute that indicates the seasons. Similarly, based on the image 5, we create an attribute that indicates whether it is a working day or not.

Based on the tables with missing and zero values, we exclude individual attributes from the Data Frame. An attribute has also been created that indicates date, so we don't need rows day, month and year separately.

```
#missing a lot of data in address, gps_coding, latitude and longtitude
df = df.drop(['address','gps_coding','latitude','longitude'],axis=1)

#we have 'date' for year, month and day
df = df.drop(['year','month','day'],axis = 1)
#we have ilumination for part of day
df = df.drop(['hour'], axis = 1)
#places with bigger population tend to have more accidents
df = df.drop(['municipality','department'],axis = 1)
```

We have seen, that during the holidays, the number of accidents drops drastically. That is why we use the table with information about the holidays and make an attribute, which suggests whether the traffic accident happened few days before or after the holidays.

```
df1 = holidays

df1.ds= df1.ds.apply(lambda x: pd.to_datetime(x))

holiday_list=[]
for i in tqdm(range(0, len(df1))):
    holiday_list.append(df1.iloc[i,0]-timedelta(days=2))
    holiday_list.append(df1.iloc[i,0]-timedelta(days=1))
    holiday_list.append(df1.iloc[i,0])
    holiday_list.append(df1.iloc[i,0]+timedelta(days=1))
    holiday_list.append(df1.iloc[i,0]+timedelta(days=2))
df["near_holiday"]=df["date"].apply(lambda x: 1 if x in holiday_list else 2)
```

Similarly, we throw out attributes with mostly missing and zero values from the table **places.csv**.

```
df_places = places

#missing a lot of data
```

```
df_places = df_places.drop(['routeNumber','routeName','roadNumber','
    homePRNumber','PRDistance'],axis=1)

#null values
df_places = df_places.drop(['centralLaneWidth','infrastructure','
    outerLaneWidth','reservedLane','schoolPoint'],axis=1)
```

For all the remaining attributes, we create graphs (image 11) to make it easier to see the distribution within each attribute (all graphs can be found in the source code, in a jupyter notebook called "visualization").

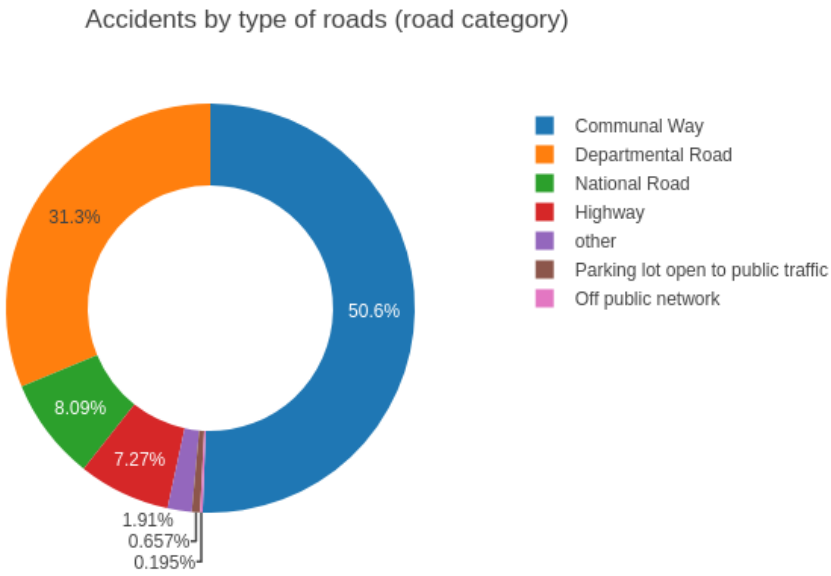


Figure 11: Types of roads on which accidents have occurred

Based on the chart, we combine several options, which are not hugely represented, into one.

```
df_places["roadCategory"]=df_places["roadCategory"].apply(lambda x: 5 if x in
    [5,6,9]  else x)

df_places["Curve"] = df_places["laneStructure"].apply(lambda x: 1 if x in
    [2,3,4] else 2)

def condition(x):
    if(x in [3,4,5,6,7,8,9]):
        return 3
    else:
        return x
#normal condtion 78%, wet 17%, null/other(oil spill, ice, mud ...) 5%
#making 3 possibilities normal, wet, other
df_places["Condition"] = df_places["surfaceCondition"].apply(lambda x:
    condition(x))
df_places = df_places.drop(['laneStructure','surfaceCondition'],axis=1)
```

```
#making on/off road
df_places["situationofAccident"] = df_places["situationofAccident"].apply(
    lambda x:2 if x in [2,3,4,5] else x)
df=pd.merge(df, df_places,on="AccidentID")
```

We connect the current Data Frame to the main Data Frame based on the accident identification number.

Finally, from the **users.csv** table, where there are not many missing values, we only exclude attributes with a large number of zero values. We observe the values of the remaining attributes and make corrections.

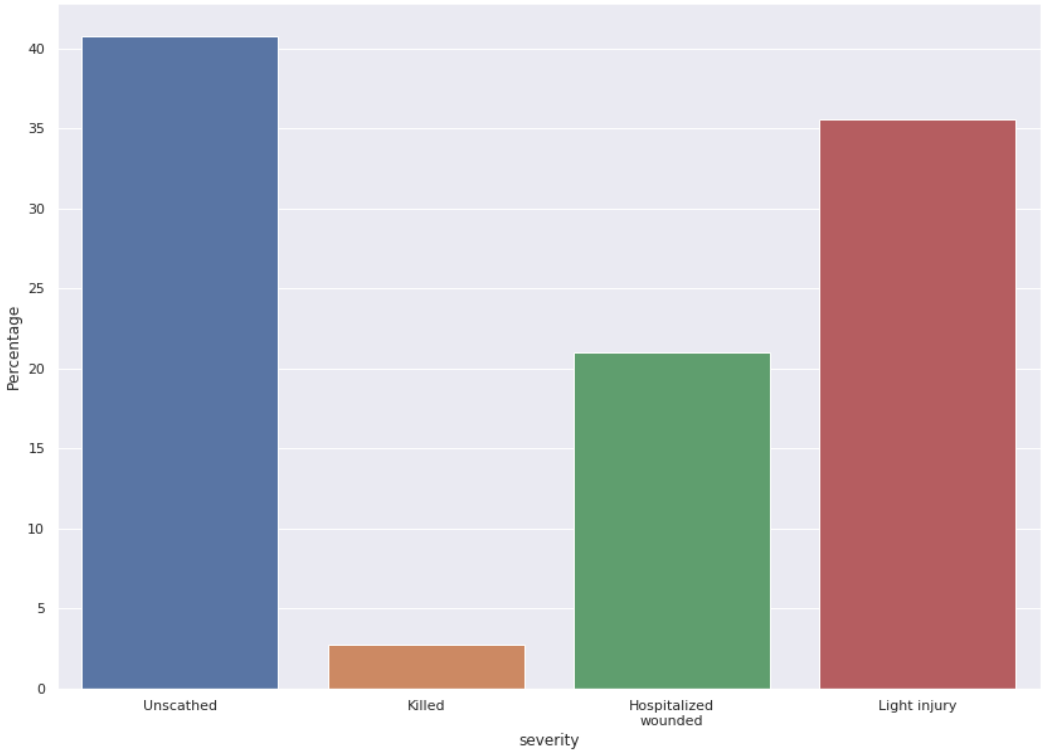


Figure 12: Consequences of the accident

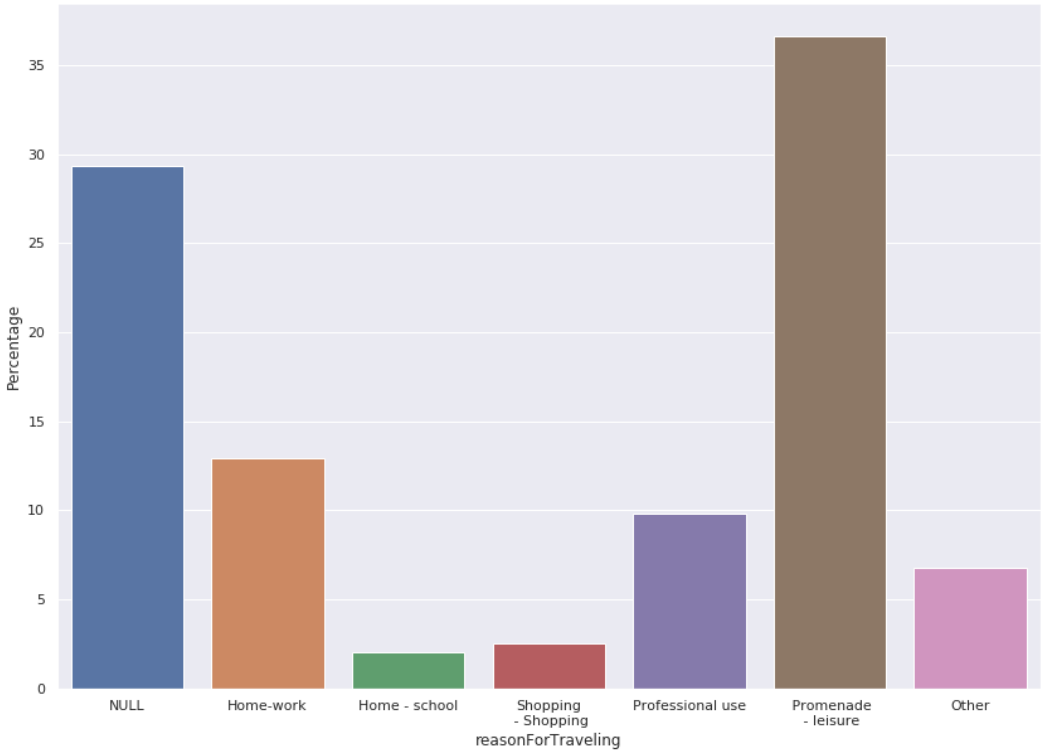


Figure 13: Reason for trip

```
df_users = users

df_users = df_users.drop(['actionOfPedestrian','locationOfPedestrian','
    pedestrianGroup'],axis = 1)

#making 3 cases
#with safety equip. 1, without 2 and unknown 3
def equipment(x):
    if(x in [11.0,21.0,31.0,41.0,91.0]):
        return 1
    elif(x in [12.0,22.0,32.0,42.0,92.0]):
        return 2
    elif(x in [13.0,23.0,33.0,43.0,93.0]):
        return 3
df_users["safetyEquipment"] = df_users['safetyEquipment'].apply(lambda x:
    equipment(x))

#value 9 is others, putting 0 values in other(no known reason for traveling)
def reason(x):
    if(x in [9,0]):
        return 6
    else:
        return x
df_users["reasonForTraveling"] = df_users["reasonForTraveling"].apply(lambda
    x: reason(x))

df=pd.merge(df, df_users,on="AccidentID")
```

The attribute, which describes the use of safety equipment, has two values, one

which describes the type of equipment and the other, which describes whether the appropriate equipment was used or not, or if there is no information. Such attribute values are impractical, so we make a correction and make an attribute with 3 possibilities (equipment used, not used or it is unknown whether it was used).

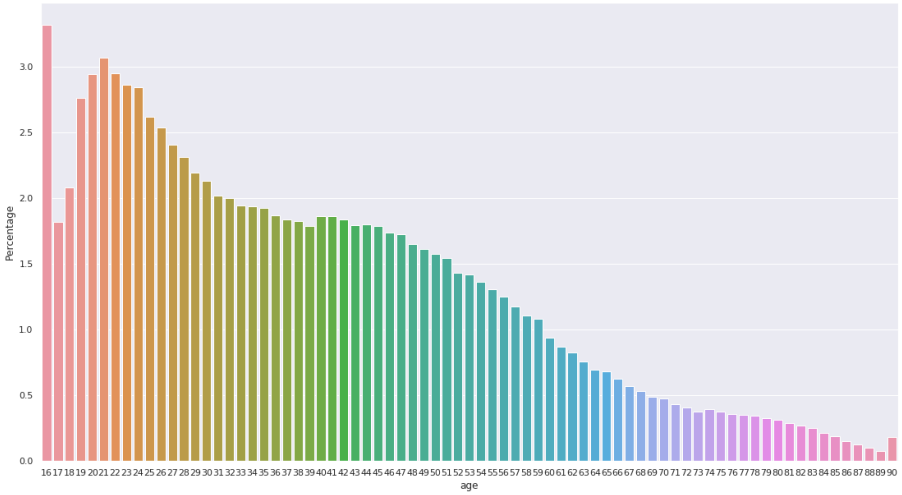


Figure 14: Driver age distribution in accidents

We make an attribute that marks the age of the person, since we see a tendency for younger drivers to participate in a larger number of accidents than more experienced ones. All drivers, over the age of 90 and under the age of 16, are merged into one group.

```
#making age atribute
df["age"] = df["AccidentID"]//100000000 - df["yearOfBirth"].astype(int)
df["age"] = df["age"].apply(lambda x: 16 if (x<17) else x)
df["age"] = df["age"].apply(lambda x: 90 if (x>89) else x)
```

3.1 Data cleaning

We check for missing and zero values in the remaining attributes.

Table 5: Remaining missing values

Attributes	Percentage
safetyEquipment	6.51%
place	5.34%
numberofLanes	0.21%
roadProfile	0.12%
yearOfBirth	0.12%
Condition	0.12%
situationofAccident	0.12%
other	<0.1%

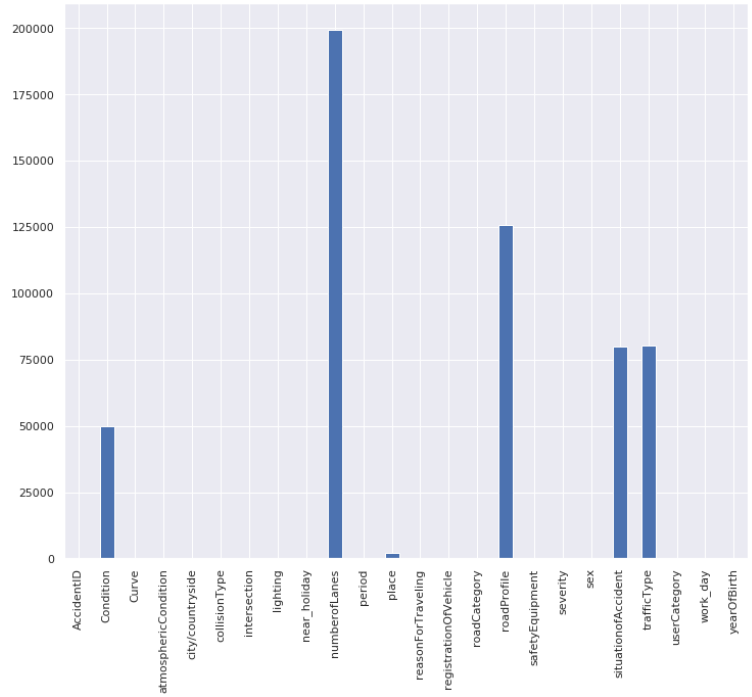


Figure 15: Zero values in remaining attributes

We make a correlation matrix and check the correlation coefficients. From the image 16 we see that the attributes place and userCategory have a high correlation coefficient (0.77). These two attributes behave similarly, the place attribute describes the position of the person in the car, and the userCategory the role of the person (for example, for the driver the position is 1 and the category is the driver).

Based on the collected data, we exclude all instances that have missing values, while the zero values are replaced with the most common value of that attribute, or an unknown value, if there is such an option ("other" if exists). We also exclude attributes for the accident identification number and vehicle registration, because all values are different, so they have no benefit in clustering.

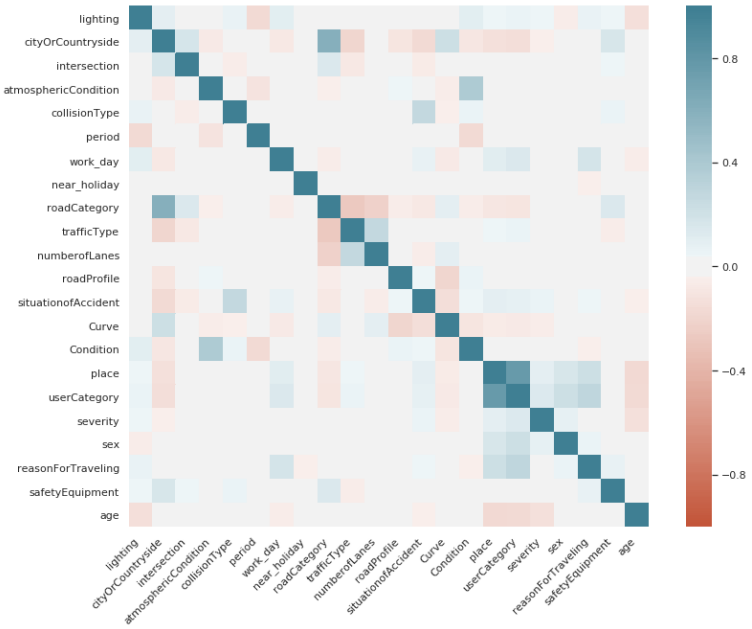


Figure 16: Correlation matrix

We also make the attribute “age”, so the year of birth is not required.

```
df = df.drop("date",axis = 1)

#drop where missing values
df = df.dropna()

#replace 0 values with most common value in pie-charts
df["Condition"] = df["Condition"].replace(0,1)
df["roadProfile"] = df["roadProfile"].replace(0,1)
df["situationofAccident"] = df["situationofAccident"].replace(0,1)
df["trafficType"] = df["trafficType"].replace(0,2)
df["safetyEquipment"] = df["safetyEquipment"].replace(0,3) #3 not known if
    safety Eq used
#number of lanes = 0 is road without central lane


avg = df["yearOfBirth"].sum()/len(df)

df["yearOfBirth"] = df["yearOfBirth"].replace(0,int(avg))

df = df.drop(["AccidentID","yearOfBirth","registrationOfVehicle"],axis = 1)

#correlation with userCategory
df = df.drop(["place"],axis = 1)
```

We download this merged table in **.csv** format and use it in algorithms.

4 K-means

First, it is necessary to determine the number of clusters. We use the elbow method (Figure 17) to determine the optimal number of clusters. For the optimal number of clusters, we pick the point where increasing the number of clusters does not bring a big gain in reducing the inertia. We cluster a data set

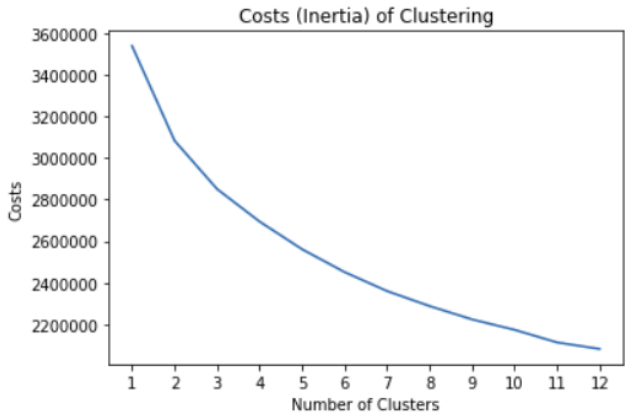


Figure 17: We will take 5 clusters based on graph

with appropriate parameters and then analyze the obtained clusters.

```
km=KMeans(n_clusters=5,init ='k-means++',n_init=20)
clusters=km.fit_predict(x)
#clustering done on whole data, name sample stayed from before when we
    clustered sampled data
sample["Cluster"]=clusters
```

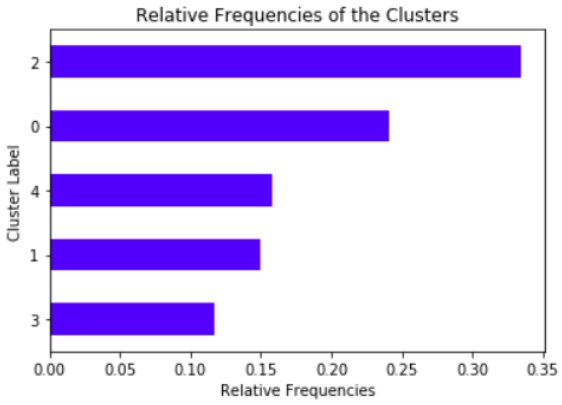
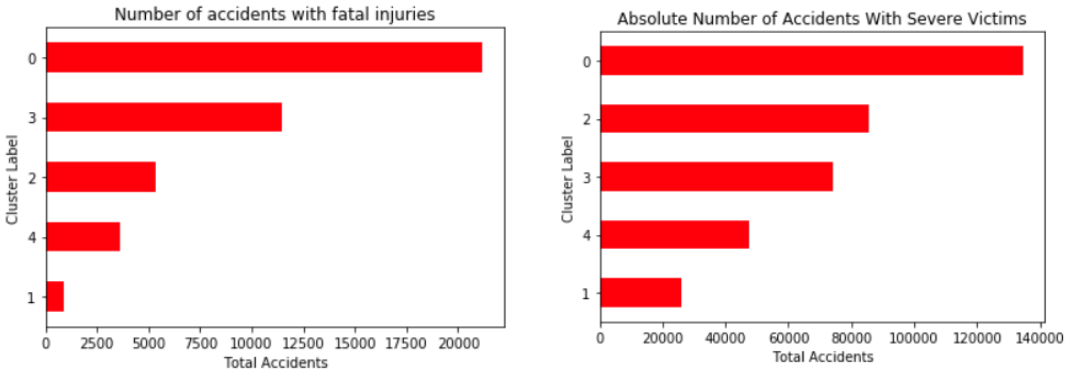


Figure 18: Frequency of resulting clusters



(a) Number of deaths in each cluster

(b) Number of injured in each cluster

Figure 19: Distribution of injuries in clusters

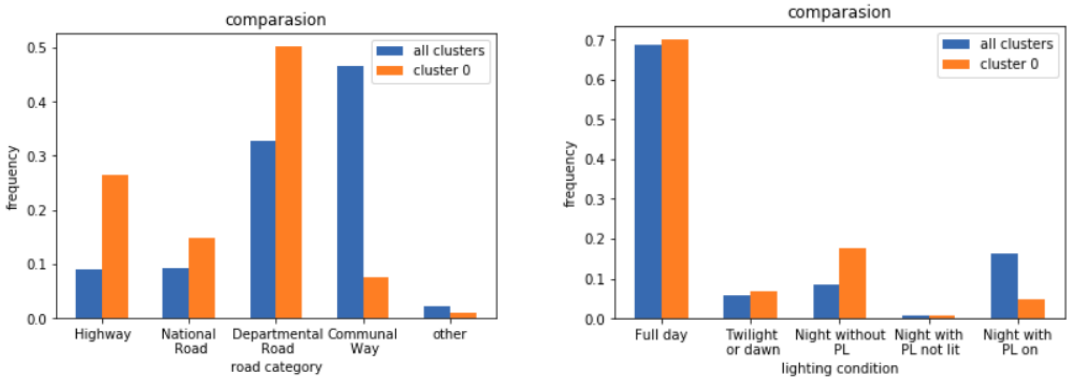
Table 6: Percentage of hospitalized with severe or fatal injuries

Cluster	**Number of accidents**	**Number of hospitalized**	**percentage**
cluster 0	406099	134850	33.21%
cluster 1	252069	26142	10.37%
cluster 2	565441	85425	15.1%
cluster 3	197030	74320	37.72%
cluster 4	267433	47552	17.78%

Table 7: Percentage of fatal cases

Cluster	**Number of accidents**	**Number of hospitalized**	**percentage**
cluster 0	406099	21192	5.21%
cluster 1	252069	913	0.36%
cluster 2	565441	5336	0.94%
cluster 3	197030	11492	5.83%
cluster 4	267433	3649	1.36%

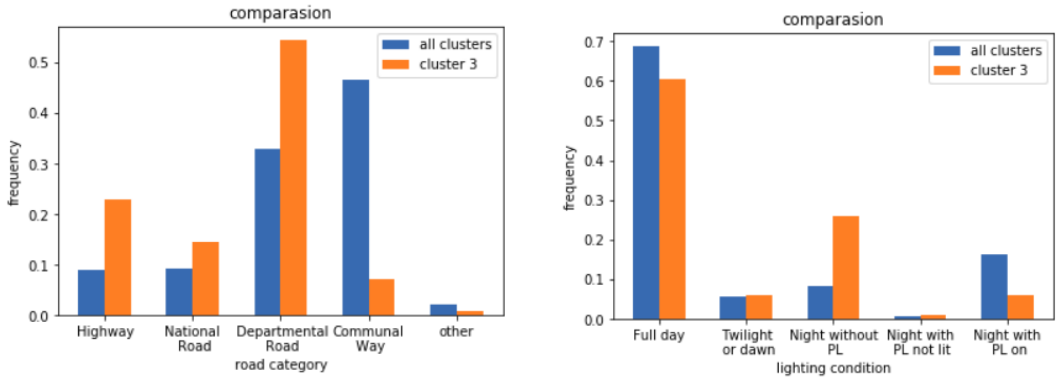
Cluster 2 is the biggest cluster (image 18). Clusters 0 and 3 are most interesting, because severe accidents occur much more often than in other clusters. We check how these clusters differ from other clusters. Below are just some of the results, which seem interesting.



(a) Type of road

(b) Brightness

Figure 20: Cluster 0 vs average



(a) Type of road

(b) Brightness

Figure 21: Cluster 3 vs average

For the observed attributes, clusters 0 and 3 show similar characteristics. Accidents in these two clusters occur in greater numbers on district and highway roads and drastically less on state roads. There is increase in accidents at night without public lighting compared to the average. From this we can conclude that driving at night, on an unlit district road, carries a greater danger. From my brief search these are mostly local routes or old national roads that have been transferred to the jurisdiction of the district.

5 K-modes

K-modes is an algorithm for clustering categorical data, which should suit better this data set. K-modes algorithm defines clusters based on the number of matches (difference vectors are made, smaller number means more similarity) between the attributes of two instances. Unlike the better known k-means algorithm, which uses Euclidean distance to cluster numerical data. More information and the source code of the algorithm can be found at link. We use the elbow method to determine the optimal number of clusters.

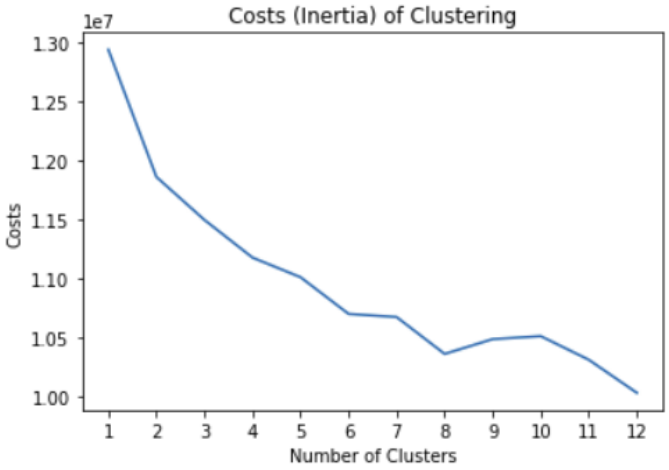


Figure 22: We will take 8 cluster base on graph

```
km=KModes(n_clusters=8,init='Huang', n_init=3)
clusters=km.fit_predict(x)
sample["Cluster"]=clusters
```

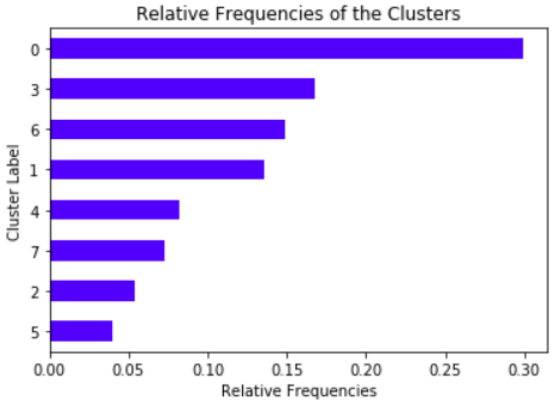
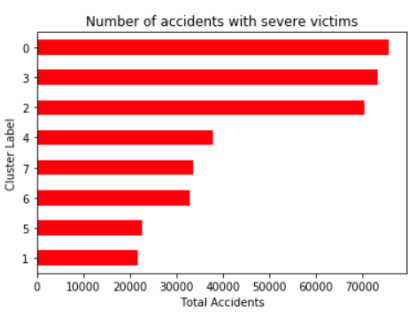
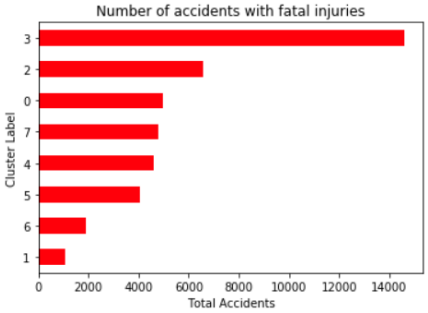


Figure 23: Frequency of resulting clusters



(a) Number of deaths in each cluster



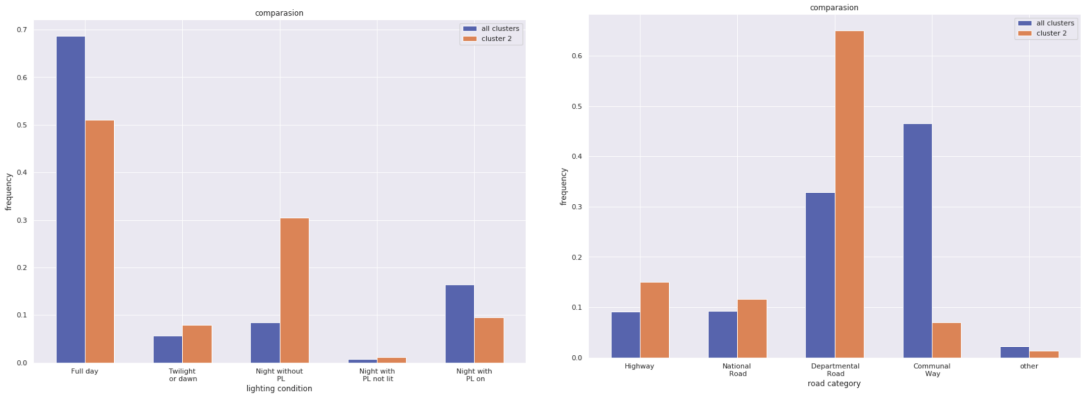
(b) Number of injured in each cluster

Table 8: Percentage of hospitalized with severe injuries

Cluster	**Number of accidents**	**Number of injured**	**percentage**
cluster 0	505915	75811	14.98%
cluster 1	228645	21782	9.52%
cluster 2	91703	70424	76.79%
cluster 3	282631	73228	25.90%
cluster 4	138043	37933	27.47%
cluster 5	66816	22580	33.79%
cluster 6	252125	32895	13.04%
cluster 7	122194	33636	27.52%

Table 9: Percentage of fatal cases

Cluster	Number of accidents	Number of deaths	percentage
cluster 0	505915	4993	0.98%
cluster 1	228645	1069	0.46%
cluster 2	91703	6581	7.17%
cluster 3	282631	14629	5.17%
cluster 4	138043	4590	3.32%
cluster 5	66816	4043	6.05%
cluster 6	252125	1878	0.74%
cluster 7	122194	4799	3.92%

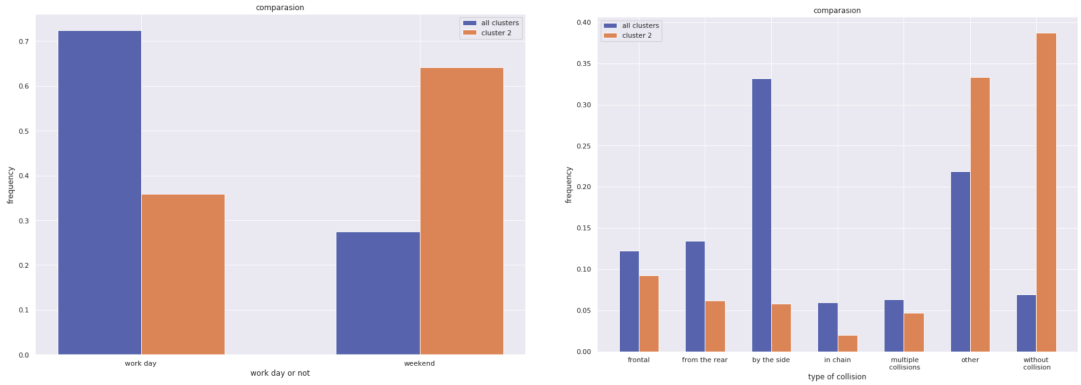


(a) Brightness

(b) Type of road

Figure 25: Cluster 2 vs average

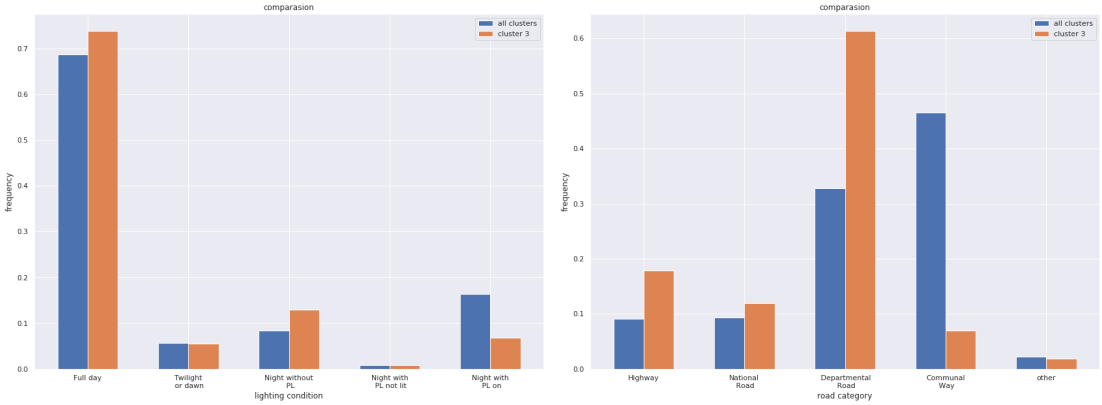
Similarly to k-means algoright, the most dangerous cluster has a higher incidence of accidents at night, on a road without lighting, which is maintained by the district. Also we can see that most accidents happen over the weekend without a collision. We can conclude that this cluster mostly contains accidents involving pedestrians, cyclists and other secondary traffic participants during the weekend, when more people go out for recreation, walks and the like.



(a) workday or weekend

(b) Type of collision

Figure 26: Cluster 2 vs average

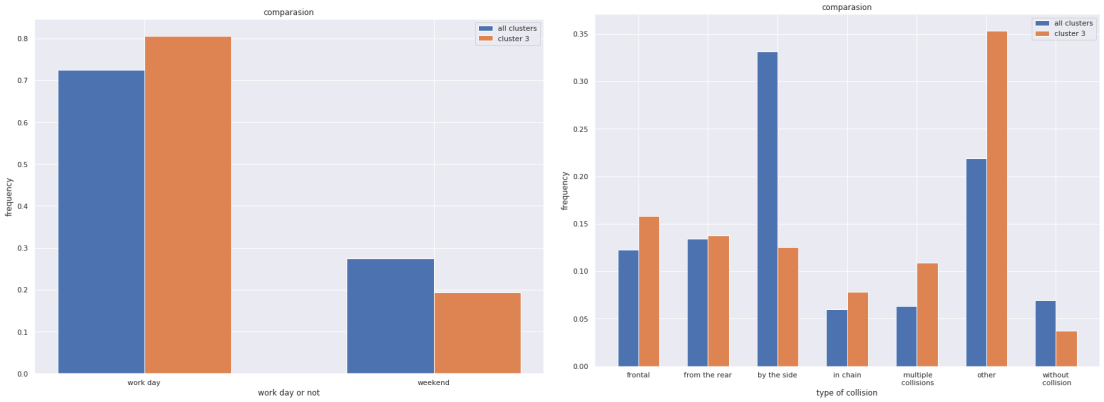


(a) Brightness

(b) Type of road

Figure 27: Cluster 3 vs average

Unlike cluster 2, in this cluster most accidents are car crashes, which occur more often on workdays than on weekends. As we have already seen in previous results, there is a tendency for collisions with greater consequences to occur on district roads. This cluster has the most deaths.



(a) Workday or weekend

(b) Type of collision

Figure 28: Cluster 3 vs average

6 Hierarchical clustering

Results presented here are results over a sample size of 20,000, unlike previous algorithms where we observed results over the entire set.
Again, we need to determine the optimal number of clusters, but this time since it is a hierarchical clustering, we use a dendrogram.

```
scaler = MinMaxScaler().fit(sample)
x = pd.DataFrame(scaler.transform(sample))

Z = linkage(x, 'ward',metric='euclidean')
fancy_dendrogram(
    Z,
    truncate_mode='lastp',
    p=30,
    leaf_rotation=90.,
    leaf_font_size=12.,
    show_contracted=True,
    annotate_above=10,
    max_d=50
)

plt.show()

from scipy.cluster.hierarchy import fcluster

sample['Cluster'] = fcluster(Z, t=50, criterion='distance')
```

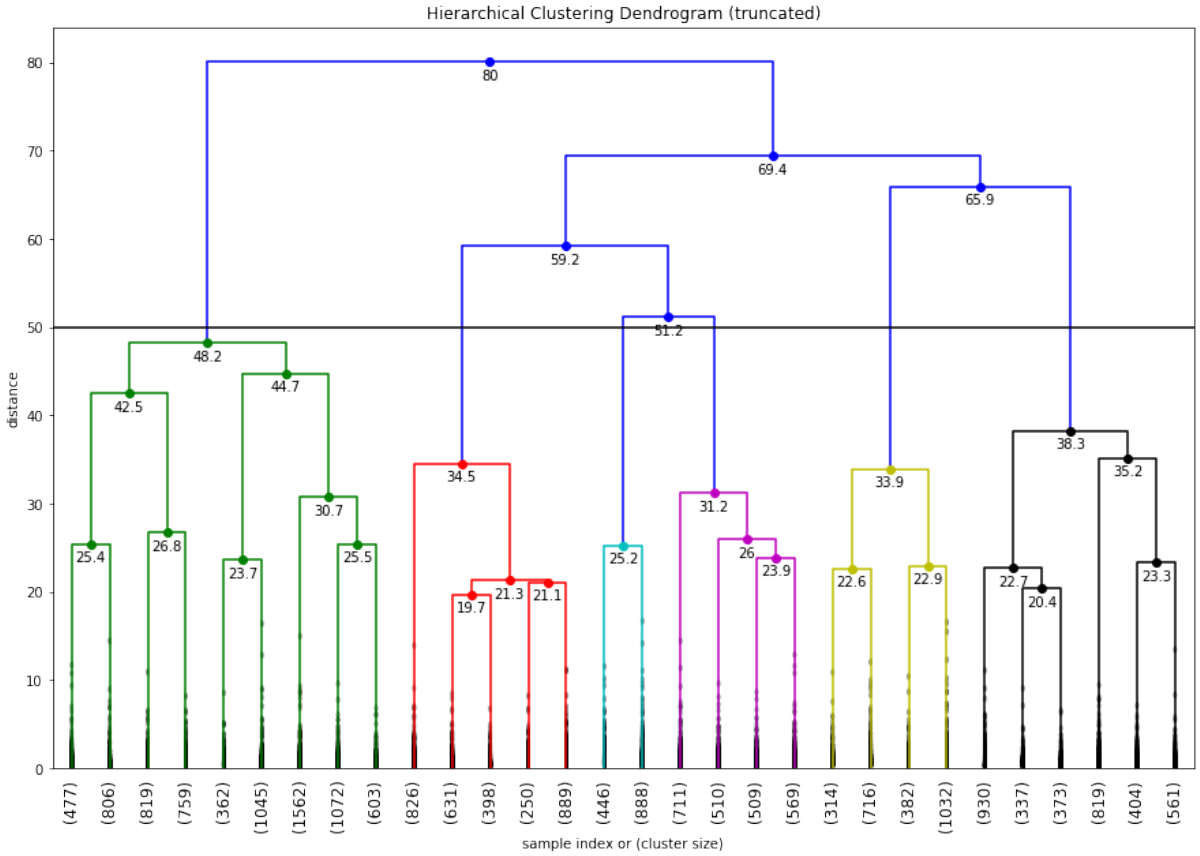


Figure 29: For distance 50, we get 6 clusters

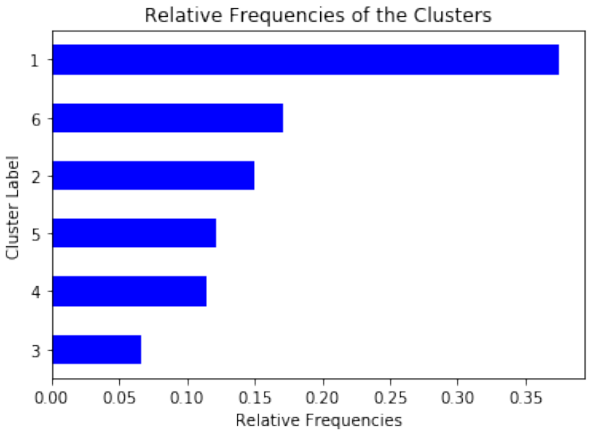
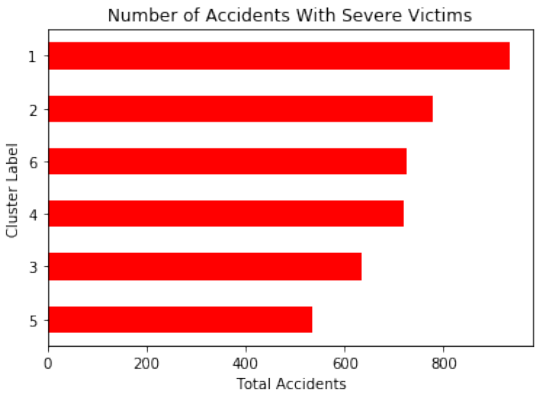
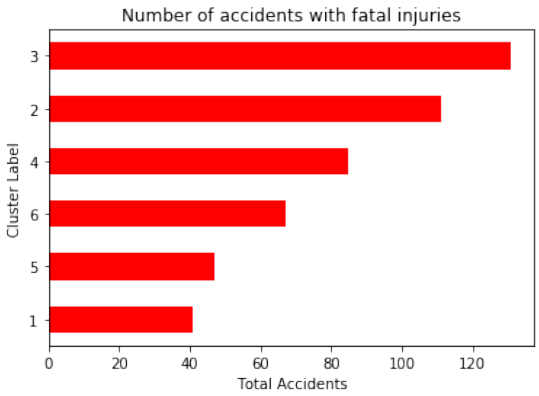


Figure 30: Frequency of resulting clusters



(a) Number of injured in each cluster



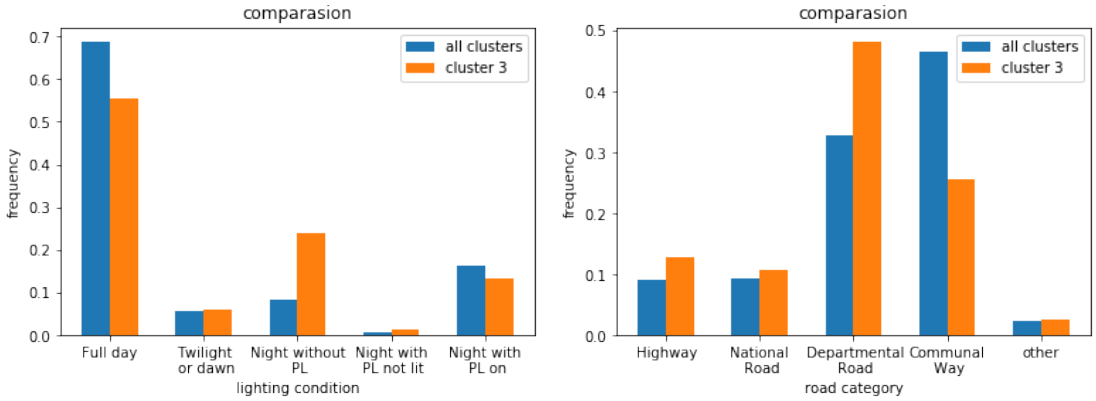
(b) Number of deaths in each cluster

Table 10: Percentage of cases with severe injruies

Cluster	**Number of accidents**	**Number of severe injuries**	**percentage**
cluster 1	7505	935	12.45%
cluster 2	2994	778	25.98%
cluster 3	1334	636	47.67%
cluster 4	2299	720	31.31%
cluster 5	2444	536	21.93%
cluster 6	3424	727	21.23%

Table 11: Percentage of fatal cases

Cluster	**Number of accidents**	**Number of deaths**	**percentage**
cluster 1	7505	41	0.54%
cluster 2	2994	111	3.70%
cluster 3	1334	131	9.82%
cluster 4	2299	85	3.69%
cluster 5	2444	47	1.92%
cluster 6	3424	67	1.95%



(a) Brightness (b) Type of road

Figure 32: Cluster 3 vs average

(a) workday or weekend (b) Type of crash

Figure 33: Cluster 3 vs average