UK road traffic prediction

Elizabeta Budini
Student ID: 19147099

Module: Advance Data Science - CMP7161

Birmingham City University
Date: 12/05/2020

Table of contents:

1. Abstract

- This study will analyse a traffic dataset from the UK government to predict the level of traffic on roads. The control of traffic has many benefits both for the government and for citizens: it can forecast traffic jams and help on building solutions against pollution.

2. Dataset description

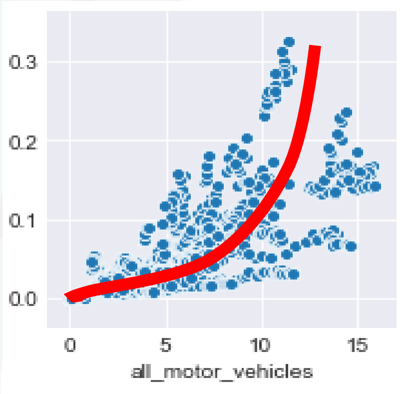
- The dataset is been published by the Department for Transport (UK) and can be found here: https://data.gov.uk/dataset/208c0e7b-353f-4e2d-8b7a-1a7118467acc/gb-road-traffic-counts
- It contains traffic level (in vehicle miles travelled*) for different vehicle types, roads and regions from 1993 to 2018.
- 14 columns and 1580 observations

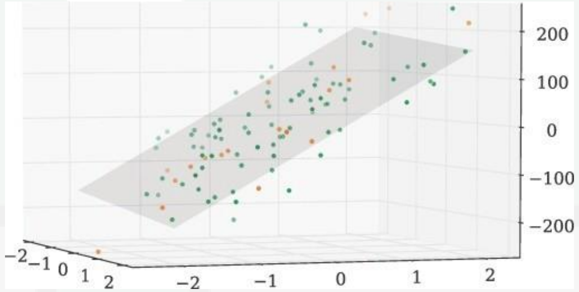
*VMT: Combination of number of vehicles in a road and travelled distance

3. Problem to be addressed

- Find a suitable ML model to predict traffic volume for all vehicle types (in billion vehicle miles) based on different features such as year, region, road category, link length, etc.

- Different techniques will be used (Linear regression, SVR, Random forest regressor)





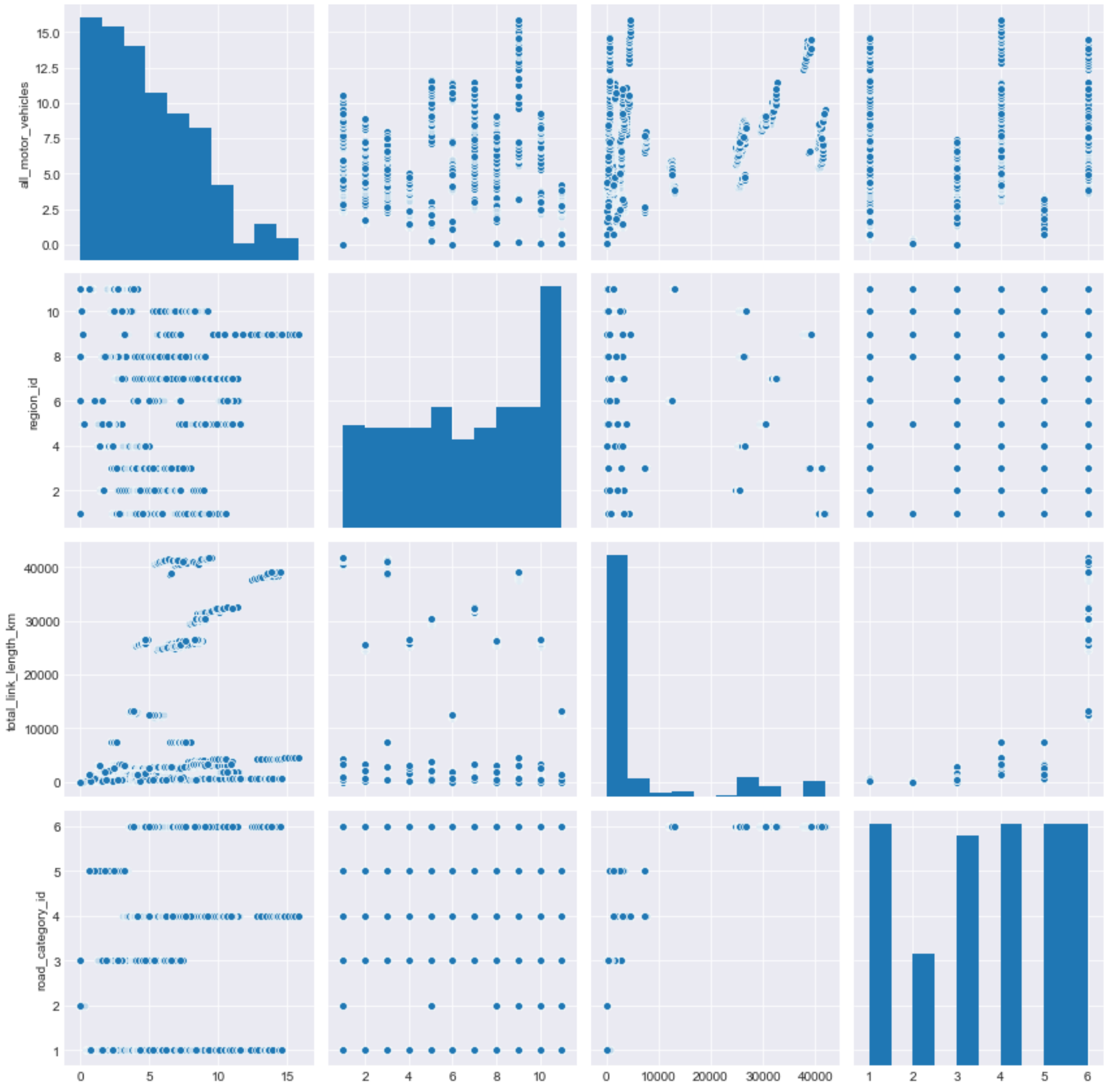
Index	year	region_id	name	ons_code	road_category_id	tal_link_length_k	al_link_length_mi	pedal_cycles	heeled_motor_ve	cars_and_taxis	uses_and_coache
1250	2014	1	South West	E12000009	2	0	0	0	0	0	0
1256	2014	2	East Midlands	E12000004	2	0	0	0	0	0	0
1262	2014	3	Scotland	S92000003	2	0	0	0	0	0	0
1268	2014	4	Wales	W92000004	2	0	0	0	0	0	0
1280	2014	6	London	E12000007	2	0	0	0	0	0	0
1281	2014	6	London	E12000007	3	0	0	0	0	0	0
1286	2014	7	East of England	E12000006	2	0	0	0	0	0	0
1316	2015	1	South West	E12000009	2	0	0	0	0	0	0
1322	2015	2	East Midlands	E12000004	2	0	0	0	0	0	0
1328	2015	3	Scotland	S92000003	2	0	0	0	0	0	0
1334	2015	4	Wales	W92000004	2	0	0	0	0	0	0
1346	2015	6	London	E12000007	2	0	0	0	0	0	0
1347	2015	6	London	E12000007	3	0	0	0	0	0	0
1352	2015	7	East of	E12000006	2	0	0	0	0	0	0

4. Pre-processing, cleaning, visualization

- There are no missing values but some road have 0 as length value
- *Region name* is a categorical feature that should be encoded, but it has the same information as *region id* column.

4. Pre-processing, cleaning, visualization

- Summarize data statistical info using describe()
- Visualize data to analyse the distribution of the features and relationships between them



5. Models description (Linear Regression)

- Approach 1: apply Linear Regression with linearly correlated features to predict *all_vehicles*
- Approach 2: apply Linear Regression with all features to predict *all_vehicles* – use RANSAC

```
Correlation with target variable
all_motor_vehicles             1.000000
cars_and_taxis                 0.997343
vans                           0.967801
two_wheeled_motor_vehicles     0.762308
buses_and_coaches              0.742106
lorries                        0.549614
pedal_cycles                   0.527216
total_link_length_miles        0.418928
total_link_length_km           0.418928
road_category_id               0.199224
year                           0.083343
region_id                     -0.059570
```

5. Models description (Linear Regression)

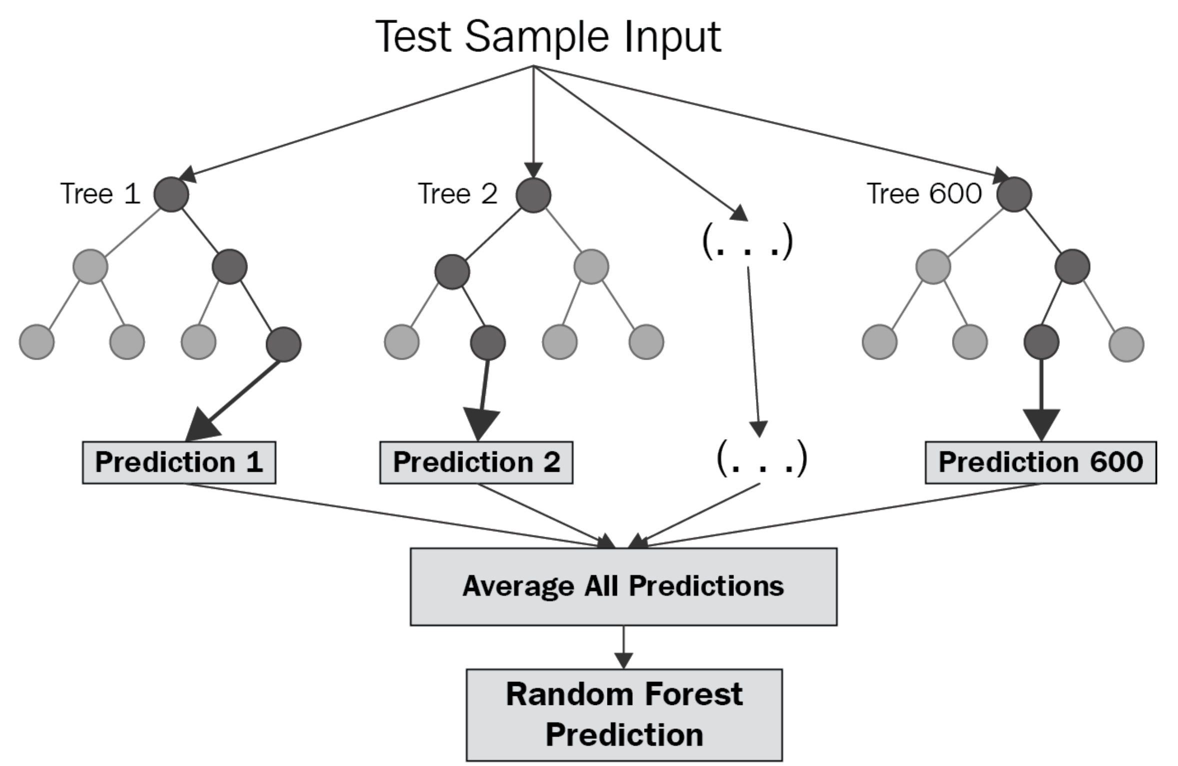
- The r2 score when considering as input linearly correlated variables (different vehicle types) is higher (99%) than the same approach with other features (20%)

```
**************************************************************
MODEL Linear regression 1
features= Index(['pedal_cycles', 'two_wheeled_motor_vehicles', 'cars_and_taxis',
       'buses_and_coaches', 'vans'],
      dtype='object')
score linear: 0.9977
regression_error: 0.1261


**************************************************************
MODEL Linear regression 2
features= Index(['year', 'region_id', 'road_category_id', 'total_link_length_km',
       'total_link_length_miles'],
      dtype='object')
score linear2: 0.2097
regression_error: 2.4114
```

5. Models description (Random Forest regression)

- Approach 1: Random forest with non-linear input features (test-train and cross-validation) and model tuning
- Approach 2: Create synthetic features to improve accuracy



frameSynthetic - DataFrame

Index	year	road_category_id	total_link_length_miles	all_motor_vehicles	SMA_5	min	max	std
0	1993	1	1950.64	41.5724	45.6714	41.5724	50.4692	3.6203
1	1994	1	1969.69	43.145	45.6714	41.5724	50.4692	3.6203
2	1995	1	1986.59	45.0765	45.6714	41.5724	50.4692	3.6203
3	1996	1	2021.21	48.094	45.6714	41.5724	50.4692	3.6203
4	1997	1	2071.12	50.4692	45.6714	41.5724	50.4692	3.6203
5	1998	1	2098.01	52.7053	47.898	43.145	52.7053	3.88302
6	1999	1	2115.44	54.0118	50.0714	45.0765	54.0118	3.58752

5. Models description (Random Forest regression)

- Use GridSearchCV, RandomizedGridSearch and Bayesian optimization to tune model hyperparameters
- Create synthetic features to improve accuracy

5. Models description (Random forest Regression)

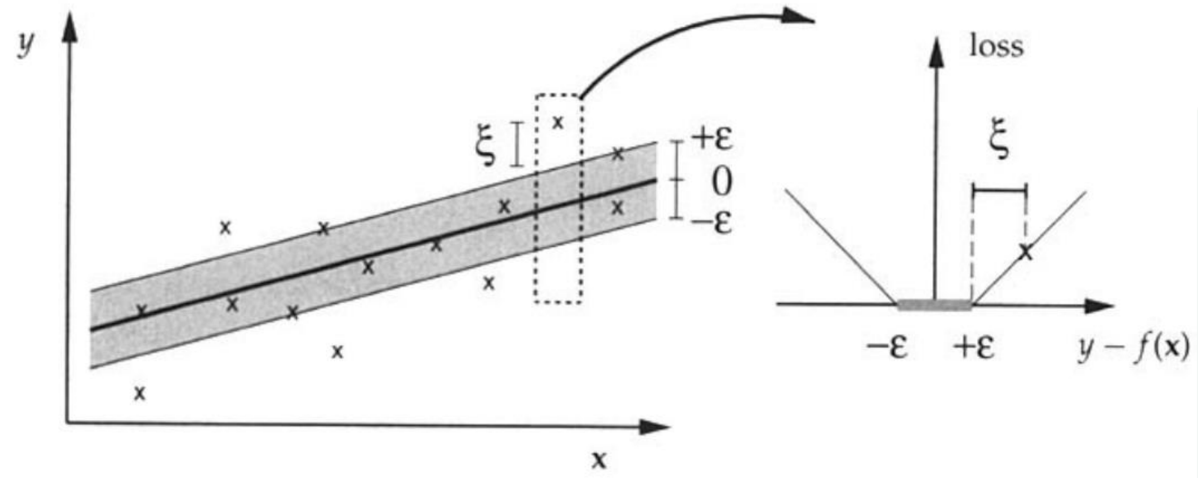
- The r2 score when considering model tuning is higher.
- Overall random forest has a high accuracy score with non-linear related features (99%)

```
************************************************************
MODEL Random forest regression with synthetic features
features= Index(['min', 'max', 'std', 'SMA_5', 'year', 'total_link_length_miles',
       'road_category_id'],
      dtype='object')
score model_forest: 0.9972
regression_error: 1.2390
```

```
************************************************************
MODEL Random forest regression hyp
features= Index(['year', 'region_id', 'road_category_id'], dtype='object')
score model_forest_hyp: 0.9977
regression_error: 0.0272
```

5. Models description (Support Vector Regression)

- Approach 1: apply SVR with GridSearchCV tuning



5. Models description (Support Vector Regression)

- GridSearchCV tuning makes SVR model too complex, to reduce execution time only one hyperparameter is tuned
- SVR accuracy score is around 67%

```
************************************************************
MODEL SV Regression
features= Index(['year', 'region_id', 'road_category_id'], dtype='object')
score: 0.6721
regression_error: 1.3146
Execution time:  2.567014455795288
```

models - DataFrame

Index	Model name	Accuracy	Error	Execution time
0	Random forest	0.9977	0.0969	8.6893
1	SVR	0.6721	1.3146	2.5670
2	Linear regression	0.1807	2.6557	0.0050

6. Results

- Mean absolute error, r2 score and execution time are compared to evaluate the best approach for each model proposed
- Linear regression is very efficient but not the best to model non-linear features.
- Random forest with model tuning achieves an excellent score in modelling non-linear features with a good execution time.

7. Conclusion and future work

- By comparing all the approaches analysed in this study, Random forest regression is recommended for modelling traffic data.
- Future work could include a Bayesian optimization for random forest instead of GridSearchCV, to better the performance in terms of execution time and control the model complexity.