INF264, Project 2: Predicting traffic

Report by Øyvind Hauge

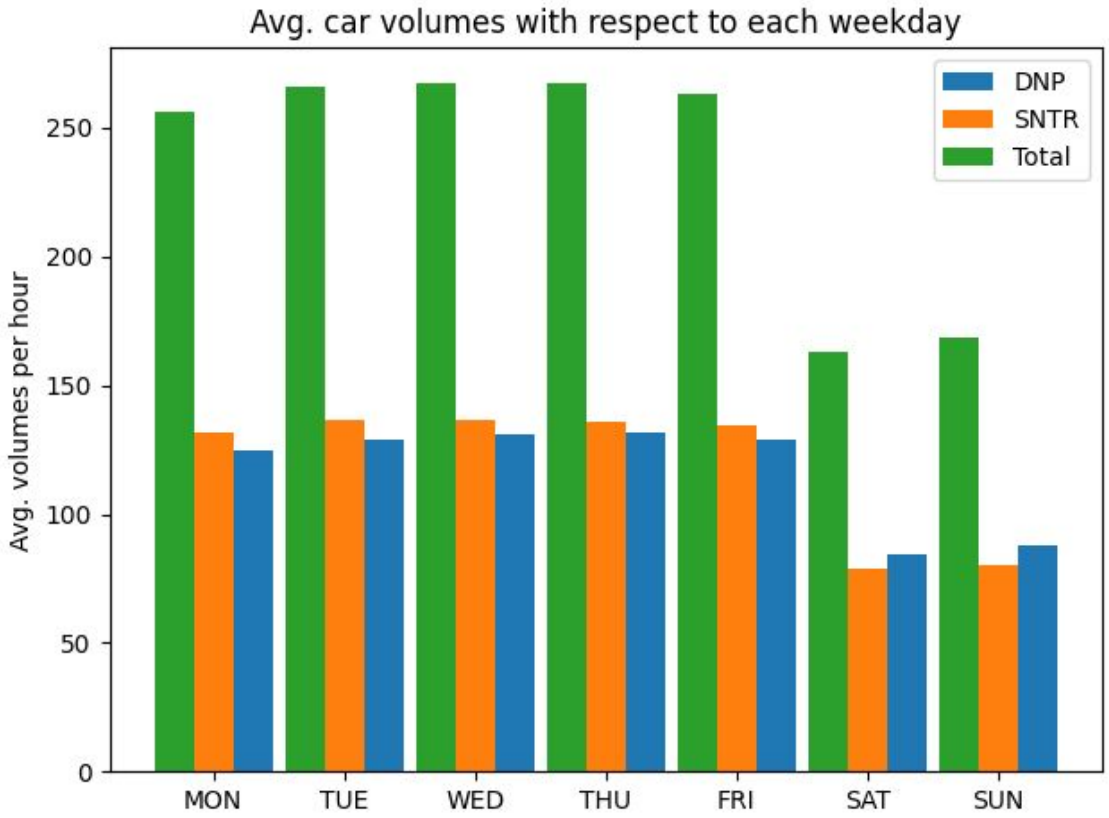
1 Report

For project 1 I wrote my code in Python (v3.8.5). In my implementation I wrap all models in a custom *Model* class that makes the mapping of the data a bit easier to work with. The rest of the implementation are utility functions for loading data and executing the models, as well as some functions used for plotting. The structure of the main file is as follows:

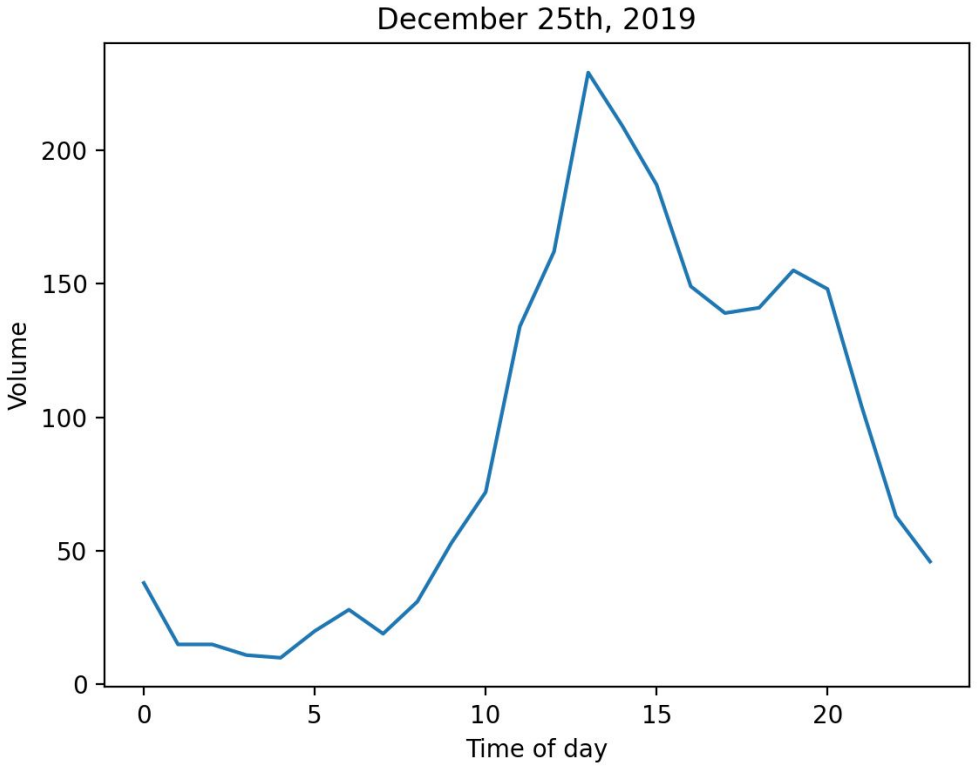
```
class Model
   -  model [the actual sklearn model that's used]
   -  name [human readable name of the model]
   -  learn() [takes data as input and learns the model]
   -  predict() [predicts the value of data points given as input]
   -  map_and_predict() [same as above but performs mapping to new features
      beforehand]
   -  score() [scores the model based on the test data]
   -  map_datapoints() [maps data points from original to new features]

load_data() [load data set from file]
preprocess_data() [creates new features based on the original data]
load_and_preprocess_data() [wrapper function]
date_from_row() [convert a DataFrame row into a datetime object]
date_is_weekend() [checks if a date is weekend]
date_is_holiday() [checks if a date is a Norwegian holiday]
date_is_gsh() [checks if a date is in the general staff holiday]
select_best_model() [selects the best model based on training acc.]
train_and_evaluate_model() [trains and scores a given model]
execute_model() [learns a model and predicts the value of unseen data]
run_instance() [runs each separate scenario - total, sntr, dnp]
```

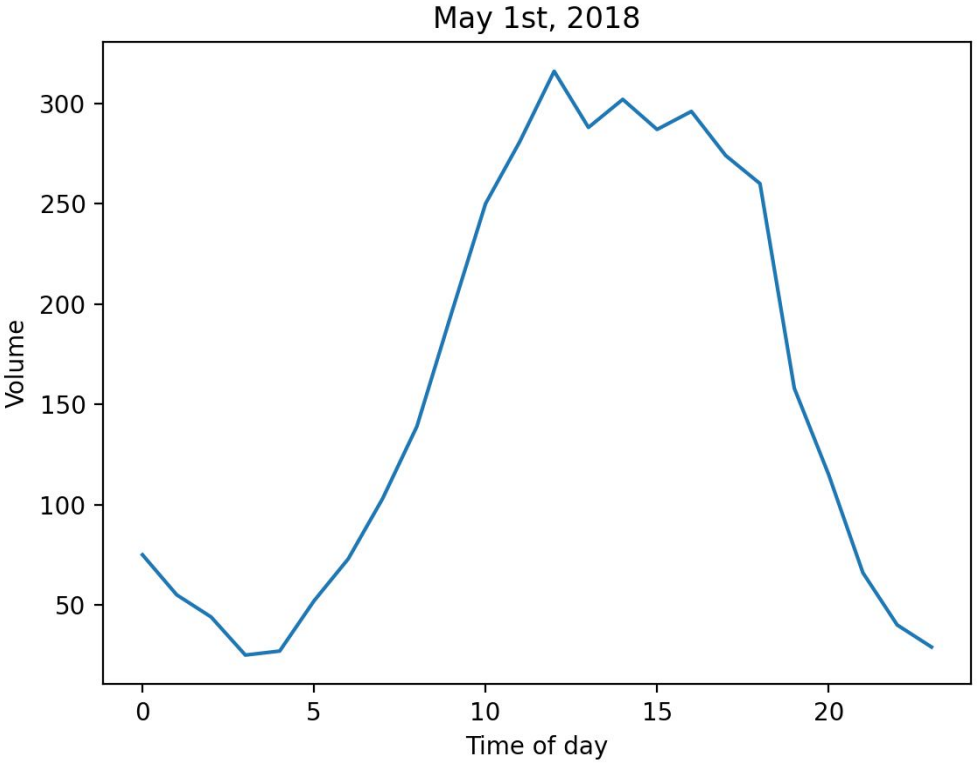
1.1 Preprocessing

Before applying the learning algorithms I did some preprocessing on the data. The first thing I did was to create a new categorical (binary) feature **is_weekend**, which separated saturdays and sundays from the rest of the weekdays. I did this because I saw a clear decrease in the average volume of cars on these days:

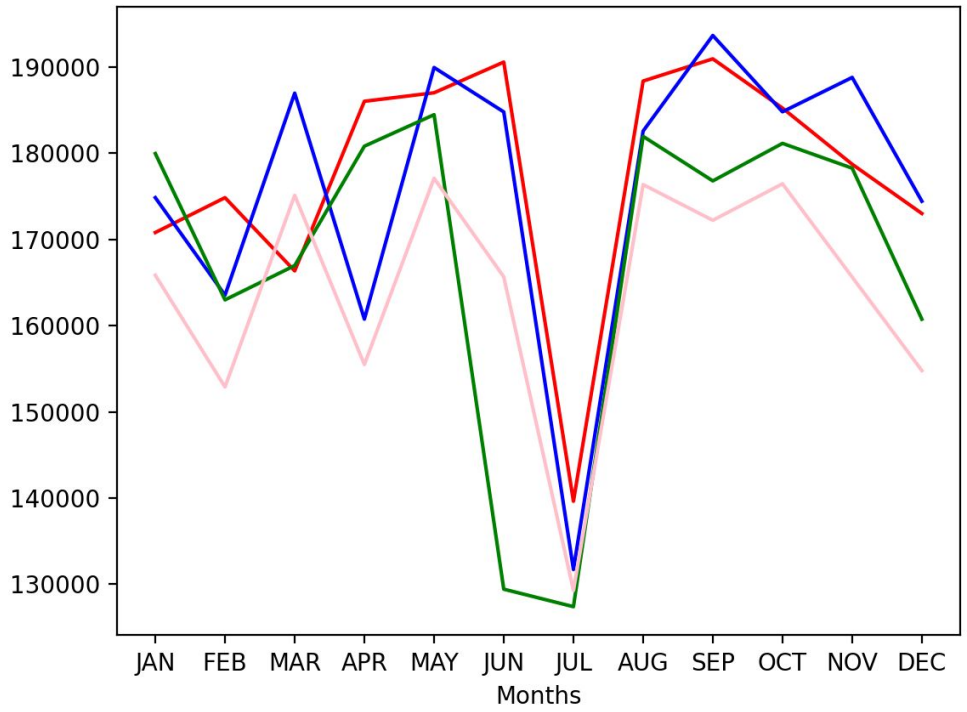


For the same reason as above, I also created a boolean feature, **is_holiday,** since I saw a consistent decrease in average car volume for days that are considered Norwegian holidays, e.g.:


December 25th, 2019
Volume
Time of day
0
50
100
150
200
0
5
10
15
20


May 1st, 2018
Volume
Time of day
50
100
150
200
250
300
0
5
10
15
20

After looking at each month in the data set, I also found that the summer months had a trend of decreasing volume, and especially the month of July, which has had a pretty significant drop in volume across all years. I then decided to use the Norwegian general holiday as a feature *("Fellesferien")* since it takes place in July. The feature is called **is_gsh** (general staff holiday).



I also used the hour column from the original data set as a feature in the new data set (**hour_of_day**) since every hour had very similar properties for pretty much every day.

1.2 Modelling and evaluation

The training accuracy below has been found by training the models with a training set and then scoring the model based in a validation set. The test accuracy is derived from unseen test data and was only calculated for the model that performed the best on the training data.

Total Volume		
Model	**Training accuracy (%)**	**Test accuracy (%)**
Support Vector Machine	0.68	-
Regression Tree	0.88	0.89

Neural Network	0.86	-

Volume to SNTR		
Model	**Training accuracy (%)**	**Test accuracy (%)**
Support Vector Machine	0.62	-
Regression Tree	0.87	0.87
Neural Network	0.83	-

Volume to DNP		
Model	**Training accuracy (%)**	**Test accuracy (%)**
Support Vector Machine	0.70	-
Regression Tree	0.85	0.86
Neural Network	0.83	-

2 Implementation

The implementation along with a README.txt and the original dataset (data.csv) can be found in the *traffic.zip* file. My implementation consists of two files. The file *traffic.py* is the main file, while *plots.py* contains different functions for plotting data. Both files can be executed in any python enabled terminal by typing *python traffic.py*.