Choose Your Own Project: Interstate Traffic Volume

edX HarvardX: PH125.9x - Data Science: Capstone

Maria Eugenia Fonseca

June 2019

Contents

1 Overview

This project is the second assignment of the 'Data Science: Capstone' course (PH125.9x), offered by edX HarvardX. The aim of this project is to use a publicly available dataset to apply machine learning techniques that go beyond standard linear regression and to clearly communicate the process and insights gained from the analysis.

1.1 Introduction

Traffic volume on a road is defined as the number of vehicles passing the measurement point per unit time. The traffic counts can be used by local councils to identify which routes are used most and to either improve that road or provide an alternative if there is an excessive amount of traffic [1].

1.2 Project Description

The objective of this project is to use machine learning models to predict the traffic volume at an American interstate and understand what features are important to explain the transit. Data transformation and feature engineering are included to improve the predictions and, as various modeling approaches are presented, the best model will be selected based on metrics such as the RMSE.

1.3 Dataset

The present project studies the Metro Interstate Traffic Volume Dataset, available at the UCI Machine Learning Repository [2]. This dataset is composed of hourly traffic volumes for westbound Interstate 94 (I-94), including weather and holiday features from 2012 to 2018.

The I-94 is an east-west Interstate Highway connecting the Great Lakes and northern Great Plains regions of the United States. Its western terminus is in Billings, Montana and its eastern terminus is in Port Huron, Michigan [3]. The measuring point for the traffic volume is roughly midway between Minneapolis and St Paul, in the state of Minnesota, as shown in the figure below [4].

Figure 1: The measuring point for the traffic volume at I-94

The dataset is downloaded directly from the UCI Machine Learning Repository. The traffic data is provided by the MN Department of Transportation, while the weather data source is OpenWeatherMap. The dataset includes the following variables:

- Response variable:
  - Traffic volume: numeric hourly traffic volume
- Features:
  - Holiday: categorical US National holidays plus regional holiday
  - Temperature: average temperature in kelvin
  - Rain: amount in mm of rain that occurred in the hour
  - Snow: amount in mm of snow that occurred in the hour
  - Clouds: percentage of cloud cover
  - Weather main: short textual description of the current weather
  - Weather description: longer textual description of the current weather
  - Date time: hour of the data collected in local CST time

Data exploration and visualization, as well as transformation and feature engineering, will be presented in the next section.

2 Methods and Analysis

2.1 Data Engineering

Before further analysis we need to engineer the data to the desired format, cleaning imputation problems, handling duplications, changing some data type to factor and creating new features.

The dataset contains 48204 hourly registers of traffic volume, weather and holiday features. The first observation is dated 2012-10-02 and the last on 2018-09-30, but between August 2014 and June 2015 there are no registers.

```
## Observations: 48,204
## Variables: 9
## $ holiday             <fct> None, None, None, None, None, None, None, ...
## $ temp                <dbl> 288.28, 289.36, 289.58, 290.13, 291.14, 29...
## $ rain_1h             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ snow_1h             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ clouds_all          <int> 40, 75, 90, 90, 75, 1, 1, 1, 20, 20, 20, 1...
## $ weather_main        <fct> Clouds, Clouds, Clouds, Clouds, Clouds, Cl...
## $ weather_description <fct> scattered clouds, broken clouds, overcast ...
## $ date_time           <fct> 2012-10-02 09:00:00, 2012-10-02 10:00:00, ...
## $ traffic_volume      <int> 5545, 4516, 4767, 5026, 4918, 5181, 5584, ...
```

The dataset presents duplicated problems. 17 observations are recorded twice, and 7629 observations are duplicated per date time (the only different features are the weather descriptions). We also have some observations described as *thunderstorm* or other rain-related description, but with 0 mm of rain that occurred in the hour. The problem occurred similarly with the snow feature.

Holiday	Temperature	Rain	Snow	Clouds	Weather main	Weather description	Date time	Traffic volume
None	275.66	0	0	90	Rain	light rain	2012-10-26 09:00:00	5234
None	275.66	0	0	90	Mist	mist	2012-10-26 09:00:00	5234
None	275.66	0	0	90	Snow	heavy snow	2012-10-26 09:00:00	5234

The duplicated observations were removed. The weather description features were kept and we fitted the model with this feature and without. The models without the variable performed slightly better, so in this report, we will only present the analysis without it.

New features were created from the original dataset: day, hour, month, year and weekday. The date 2016-12-26 was mislabeled classified as Christmas Day, so we changed it to 2016-12-25. Besides, when analyzing the entire holiday variable, we noticed that a holiday was only labeled as such on its first hour. For instance, 2012-12-25 00:00:00 is labeled *Christmas Day*, but 2012-12-25 01:00:00 and all the other following hours of

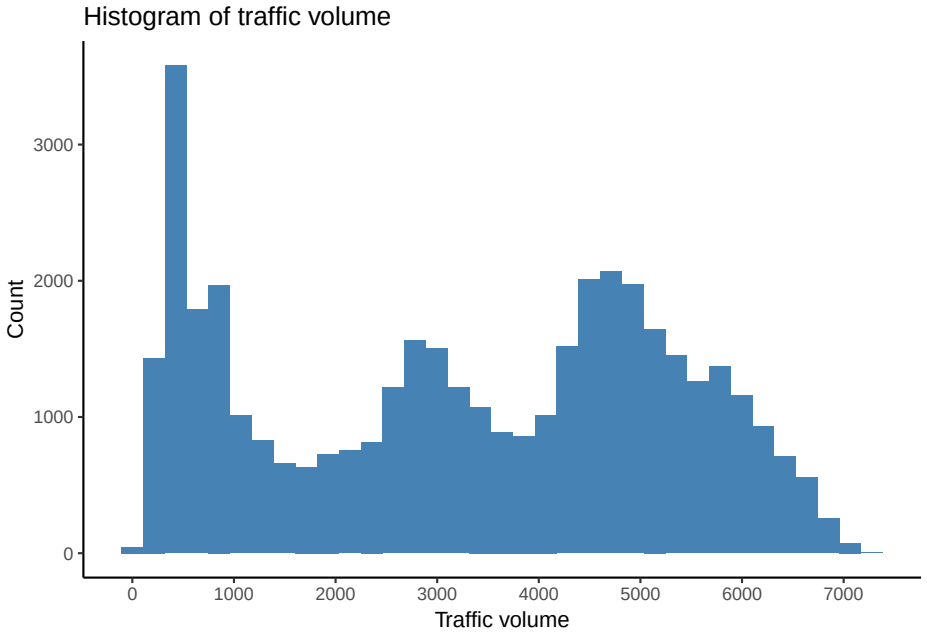
the mentioned day are labeled as *none*. We corrected this in the dataset and moved from 61 observations labeled as holidays to 1203. We created a binary variable that acknowledges if a day is or not a holiday and two other binary variables that notice if the previous or following day is a holiday. We noticed a few observations with more than one measure of temperature, rain or cloud cover; in these cases, we calculated the mean of the observed values.

Ten observations had a registered temperature of 0 Kelvin, which is not possible as it has never been observed on Earth. In like manner, a rain of 9831.3 mm per hour was observed, when the record registered is a much lower 305 mm/hour. All eleven observations were removed from the dataset.

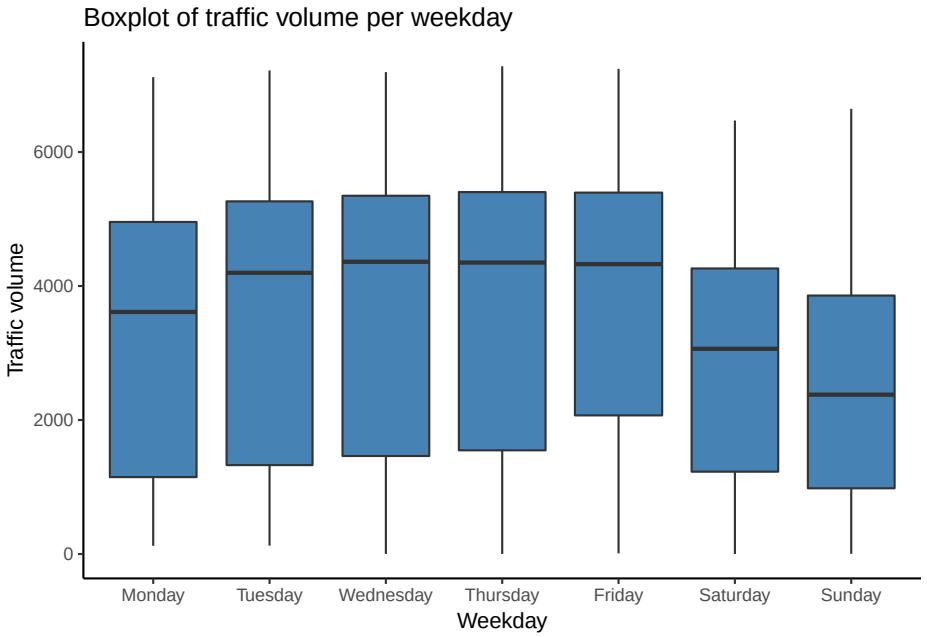
2.2 Exploratory Data Analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations [5].

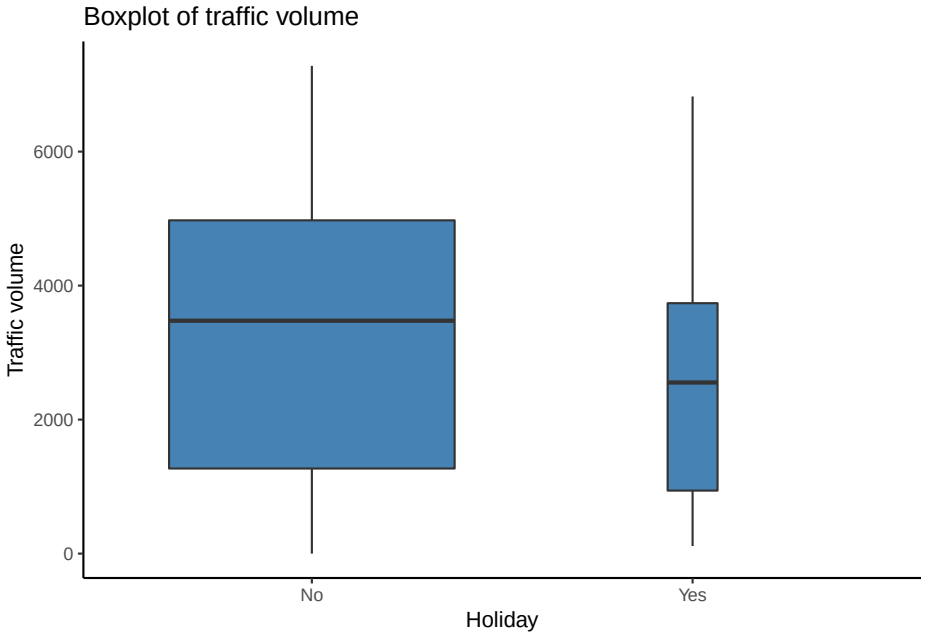
The traffic volume follows a multimodal distribution with three peaks. The first peak contains the highest frequency of traffic value, below 1000 vehicles per hour. The second peak occurs around 3000 vehicles/hour and the third peak, around 4500.



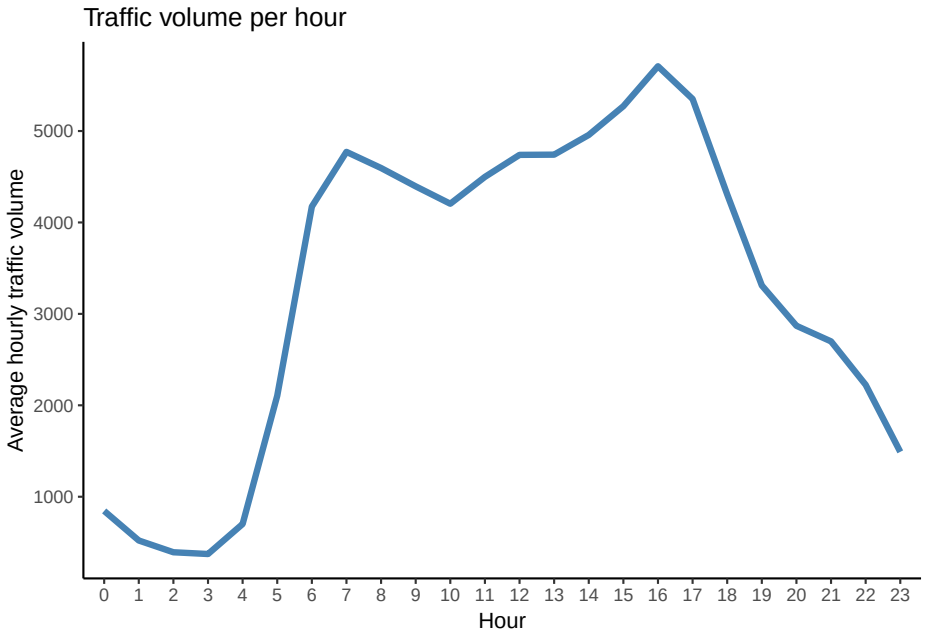
Some new features were created from the original dataset, such as the weekday. As shown in the boxplot below, the traffic volume appears to increase slowly over the weekdays and is considerably lower on weekends.



As presented in the boxplot below, the traffic volume appears to be slightly lower during the holidays.



The traffic volume varies per hour of the day, which is indicative that it will be a good feature of the predictive model. The first big peak in the traffic volume of the day is early in the morning, from 6 to 7 am. The traffic decreases slightly in the late hours of the morning, but increases again after lunch, reaching its maximum between 4 and 5 pm.



In machine learning is important to consider two different datasets - the training dataset and the test dataset. The training dataset is the sample of data used to fit an algorithm, and the test dataset is the sample of data used to provide an unbiased evaluation of the final model fit. For this study, our train set will be all the data until 2017 (34031 observations) and our test set will be data of the year 2018 (6533 observations).

As only 31 observations had snow registered, and none in the last year (test data) this feature will not be included. To proceed to modeling, the final features considered were holiday, previous or following day of a holiday, temperature, percentage of cloud cover, hour of the day and weekday.

2.3 Modeling Approaches

To guide the process of choosing the machine learning model, we did some experimenting with a subset of the data. We modeled 1-year of data with algorithms such as elastic net, bagged tree and SVM and choose Caret's eXtreme Gradient Boosting (xgbTree) because of a tradeoff between RMSE and execution time. In order not to overstretch this report as some of those models took a long time to run, the experimentation part was omitted and we will focus on tunning the boosting model.

2.3.1 Linear model and xgbTree model with default hyperparameters

To provide a reference point we set up two baseline models: a simple linear regression model and the xgbTree model with default hyperparameters. This is to see what kind of effect the tuning has on the model performance. All fits were modeled with the same train control, performing cross-validation with 3 folds.

The eXtreme Gradient Boosting (xgbTree) model has seven tuning parameters: number of boosting iterations, maximum tree depth, shrinkage, gamma - minimum loss reduction, subsample ratio of columns, minimum sum of instance weight and subsample percentage. We will change these tuning parameters in steps so that our grid is not too big.

Predictor	RMSE (train)	R squared (train)	Time (secs)
Linear model	782.98	0.84	6.00
xgbTree - Default	527.57	0.92	166.18

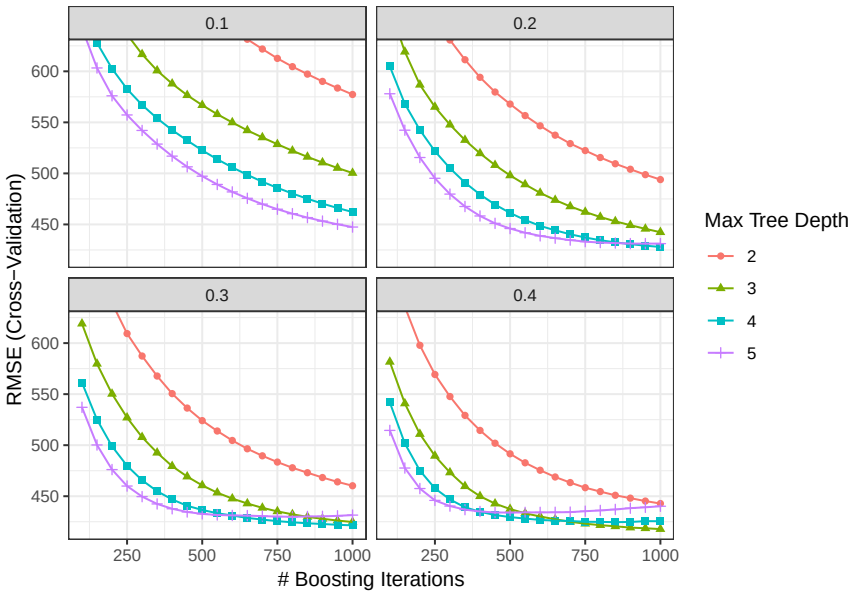
The default eXtreme Gradient Boosting already improved substancially the RMSE compared to the linear

regression (from 782.98 to 527.57 on the train set). The default model had the following tuning parameters: nrounds = 150, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 0.5.

To get reasonable running time while testing hyperparameter combinations with *caret* we don't want to go over 1000 in the number of boosting iterations. After tuning the other parameters we will come back.

2.3.2 xgbTree model Step 1: Number of iterations and the learning Rate

For the first step, we created a grid search with different boosting iterations, shrinkage and max tree depth.

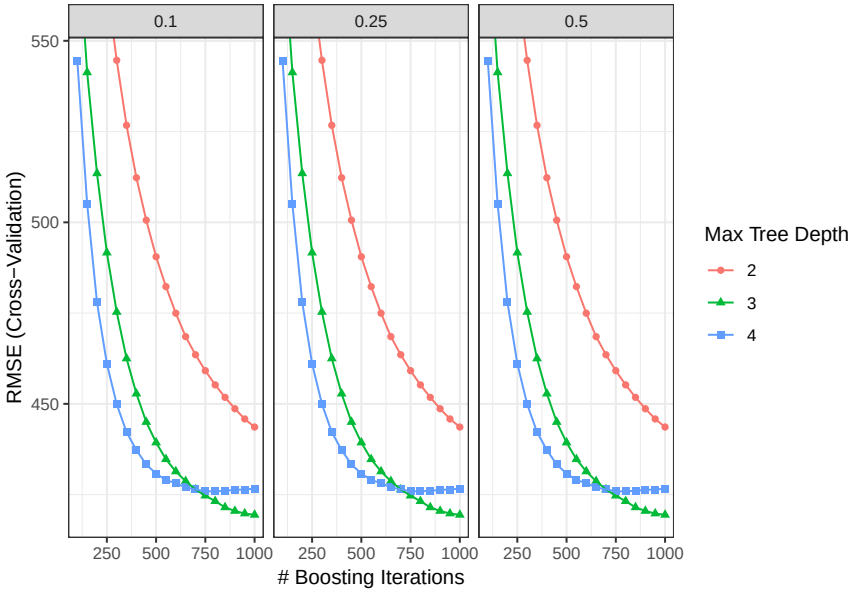


Predictor	RMSE (train)	R squared (train)	Time (secs)
Linear model	782.98	0.84	6.00
xgbTree - Default	527.57	0.92	166.18
xgbTree - Step 1	370.38	0.95	888.74

The best model found within the grid search had the tuning parameters nrounds = 1000, max_depth = 3 and eta = 0.4. As shown in the graph above, for lower shrinkage the model does not seem stable. This first tuning already improved the RMSE considerably, from 527.57 to 370.38 (-29.8%).

2.3.3 xgbTree model Step 2: Maximum Depth and Minimum Child Weight

Now we will fix the shrinkage to the optimal value found and perform a grid search on the minimum child weight and the maximum tree depth to 3 +- 1 (one above and one below the suggested best tune found in the previous step).

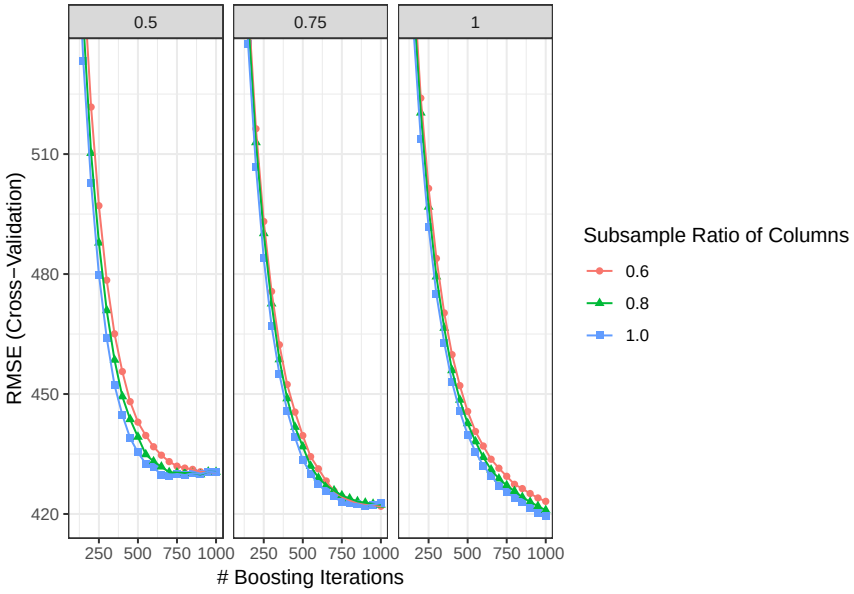


Predictor	RMSE (train)	R squared (train)	Time (secs)
Linear model	782.98	0.84	6.00
xgbTree - Default	527.57	0.92	166.18
xgbTree - Step 1	370.38	0.95	888.74
xgbTree - Step 2	370.38	0.95	475.17

The best model found within the grid search had the tuning parameters max_depth = 3 and min_child_weight = 0.1. The RMSE did not change with the different minimum child weight.

2.3.4 xgbTree model Step 3: Subsample ratio of columns and subsample percentage

In the next step, we fix the minimum child weight to the optimal value found previously, set the maximum tree depth to 3 and do a grid search on the subsample ratio of columns and subsample percentage.



Predictor	RMSE (train)	R squared (train)	Time (secs)
Linear model	782.98	0.84	6.00
xgbTree - Default	527.57	0.92	166.18
xgbTree - Step 1	370.38	0.95	888.74
xgbTree - Step 2	370.38	0.95	475.17
xgbTree - Step 3	370.38	0.96	440.20

The best model found within the grid search had the tuning parameters colsample_bytree = 1 and subsample = 1, the same used in step 2. Because fo this, the RMSE is the same.

2.3.5 xgbTree model Step 4: Gamma

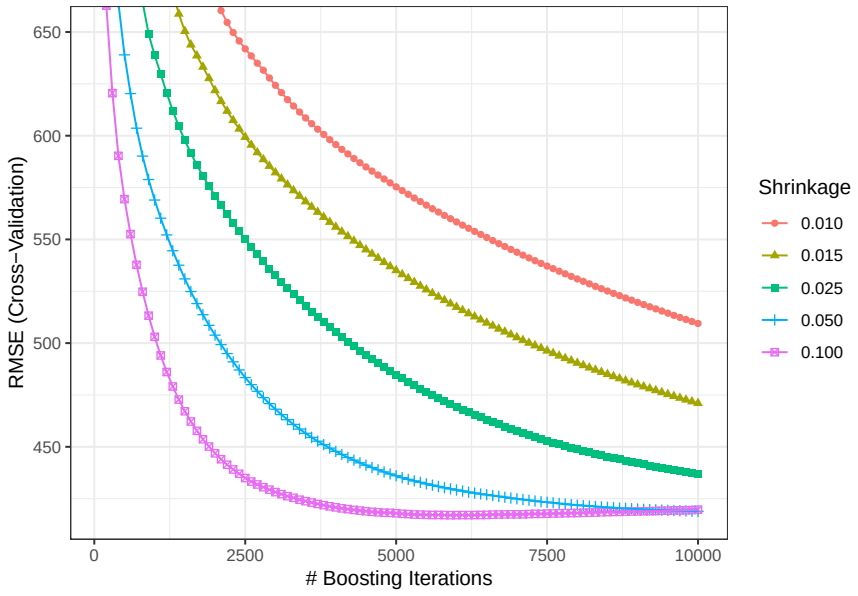
Now we will fix the colsample_bytree and subsamples tuning parameters and perform a grid search on gamma (minimum loss reduction parameter).

Predictor	RMSE (train)	R squared (train)	Time (secs)
Linear model	782.98	0.84	6.00
xgbTree - Default	527.57	0.92	166.18
xgbTree - Step 1	370.38	0.95	888.74
xgbTree - Step 2	370.38	0.95	475.17
xgbTree - Step 3	370.38	0.96	440.20
xgbTree - Step 4	370.38	0.93	335.47

Different gamma values did not have any effect on the model fit (RMSE), so we continue with the previous value.

2.3.6 xgbTree model Step 5: Reducing the Learning Rate

Now that we have tunned all hyperparameters parameters, we can go back and try different values for the number of boosting iterations and shrinkage. Before, we tried up until 1000 iterations to save running time, but now the grid search executes up to 10000 iterations.



Predictor	RMSE (train)	R squared (train)	Time (secs)
Linear model	782.98	0.84	6.00
xgbTree - Default	527.57	0.92	166.18
xgbTree - Step 1	370.38	0.95	888.74
xgbTree - Step 2	370.38	0.95	475.17
xgbTree - Step 3	370.38	0.96	440.20
xgbTree - Step 4	370.38	0.93	335.47
xgbTree - Step 5	360.85	0.96	3387.39

3 Results

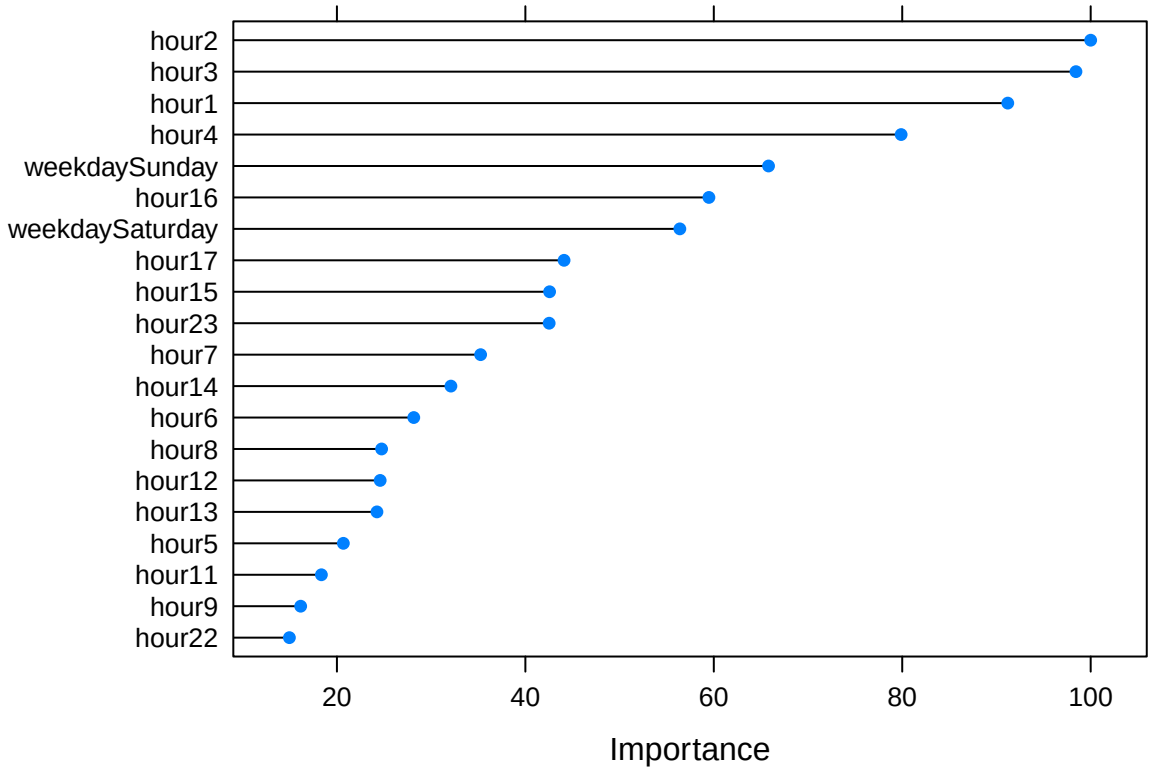
The final model had the following tuning parameters:

nrounds	eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample
5900	0.1	3	0	1	0.1	1

Now we will evaluate the model fit on the test set. A good model that does not overfit performs similarly on both train and test sets. It is expected for the RMSE to increase and, as shown in the table bellow, the RMSE on the train dataset is 360.85 and 379.44 on the test set. Even with the inflation, the test RMSE still indicates a good fit and is also better than the train RMSE of the xgbTree model with default hyperparameters.

Predictor	RMSE (train)	RMSE (test)	R squared (train)
xgbTree - Final model	360.85	379.44	0.96

Besides modeling, it is also important to understand what is relevant to explain the traffic volume. From the graph below, we can see the 20 most important variables. Hour is the most significant feature to explain transit, and the days Sunday and Saturday are also relevant, pointing out that the traffic does differ on weekends.



4 Conclusion

The objective of this project was to use machine learning models to predict the traffic volume at an American interstate and understand what features were important to explain the transit. In this report, we tried many models, created some new features, selected Caret's eXtreme Gradient Boosting (xgbTree) because of a tradeoff between RMSE and execution time and tunned its seven different hyperparameters. The final model obtained an RMSE of 360.85 in the training set and 379.44 in the test set. The most important feature to explain traffic *hour*.

5 References

https://en.wikipedia.org/wiki/Traffic_count

https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume#

https://en.wikipedia.org/wiki/Interstate_94

https://github.com/dreyco676/Anomaly_Detection_A_to_Z/blob/master/Anomaly%20Detection%20A%20to%20Z.pptx

https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

https://www.kaggle.com/pelkoja/visual-xgboost-tuning-with-caret