

# How To Best Determine Scheduled Block Times For A Dutch Airline

David Lotten<sup>1</sup>

Vrije Universiteit, Amsterdam, postcode, NL,  
d.lotten@student.vu.nl

**Abstract.** Airlines estimate block times almost a year in advance, as it is an essential building block of the season’s schedule, and to timely acquire airport slots. Accurate estimation of block times is important for customer satisfaction, slot retention and optimal deployment of crew and aircraft. Like most airlines, Dutch airline Transavia estimates block times selecting a quantile of the previous season’s actual block times per route. Based on data that is available a year in advance this paper investigates whether the performance of the current method can be improved up on and shows under what circumstances taking a quantile of historical block times is a robust method to determine future block times.

**Keywords:** Block times, Predict, Airline, Quantile

## 1 Introduction

Airlines need to estimate the block times for all their Origin / Destination pairs almost a year in advance. The time it takes to get from an Origin airport to a Destination airport is input to determine an optimal (good) roster for the crew and the fleet of aircraft. Also airport slots need to be secured on time and schedules need to be published well in advance. The definition of block time according to the Eurocontrol lexicon, [https://ext.eurocontrol.int/lexicon/index.php/Flight\\_time](https://ext.eurocontrol.int/lexicon/index.php/Flight_time), is found under flight time: “The total time from the moment an aircraft first moves under its own power for the purpose of taking off until the moment it comes to rest at the end of the flight”, pointing out in note 1 “Flight time as here defined is synonymous with the term ’block-to-block’ time”.

The stochastic nature of the block times on airline operations has been subject of many studies. Hansen and Hao [1] (2013) use data made public by the US Department Of Transportation, to deduce the policies of how different US based airlines determine and publish their block times. Sohoni et al (2011)[5] state “the schedules ignore block-time uncertainty (variance) and hence fail to capture robustness measures”. Litvay (2012) [3] also uses DOT data to design and test a novel way of determining next season’s block times. Litvay proposes as a better solution by taking the mean of the block time of the previous season, grouped by month, day of the week and he divides the day in to three blocks: departure times until 10:00h in the morning and from 10:00h in the

morning until 20:00h in order to capture congestion in the prediction. Maier and Sinai [2] also use DOT data to explain their title “Why Do Airlines Systematically Schedule Their Flights to Arrive Late?”. All these papers have three things in common: they use publicly available data, are about US airlines and are relatively outdated. When observing European low cost carriers arrival on time performance (OTP) at [urlhttp://www.flightstats.com/company/monthly-performance-reports/airlines/](http://www.flightstats.com/company/monthly-performance-reports/airlines/) these are closer to 80% than more late than not. One important difference may be the strict EU261 regulation “Regulation (EC) No 261/2004 of the European Parliament and of the Council of 11 February 2004 ...” which give passengers the right to a compensation often several times as high as the ticket price, when they arrive more than 3 hours late at their destination.

This paper consists of two parts. In the first part the circumstances under which taking the current method of taking a percentile performs well, are explained. In the second part simple statistical tests, linear regression and a tree based model will be applied to actual historical Transavia flight data in an attempt to find patterns and improve upon the performance of the current method of determining next season’s SBT.

Some basic definitions and practices first need to be established and the impact of not arriving as scheduled clarified.

### 1.1 Block time definition and composing parts

The block time consists of three parts, taxi out time, flight time and taxi in time.

**Taxi out time** is the time it takes from the literal removal of big rubber blocks in front and behind the wheels of the aircraft to keep it from rolling away unintendedly, after which the aircraft is pushed back (aircraft can not move backwards under their own power, they need a ”push back”) from the gate or stand to the runway, including deicing procedures.

**Flight time** is the time from the moment the plane is “airborne” until “touch down”: the time spent in the air.

**Taxi-in time** is the time that elapses from ”touch down” on the runway to taxiing to the gate or stand at the destination airport, where again the big rubber blocks are placed around the wheels, keeping the aircraft in place.

### 1.2 Influences on block time components

Taxi times are influenced by the distance from the gate or stand to the runway: at Amsterdam’s Schiphol Airport taxiing to and from the “polderbaan” may take up to 18 or 23 minutes depending on the “Zwanenburgbaan” being in use

or not, while the mean taxi out time is only 13 minutes. Wind directions at the flight level may influence the flight time. The pilot may influence the flight time by pushing down on the throttle, though for the distances Transavia operates on, these differences are small. Also when delayed, the pilot may be asked local ATC for a direct route, which may be granted, for example, when military airspace that otherwise needs to be circumnavigated is not in use. Aircraft do not fly in straight lines, but from way point to way point. The maximum distance reduction by flying a direct route, may differ from route to route. Also thunderstorms that increase after take off need to be circumnavigated to avoid lightning strikes. Like most airlines, Transavia starts creating schedules almost a year in advance, in order to timely involve internal stakeholders and to secure airport slots on time. None of the above described factors causing randomness is available at that time and cannot be used as input for models.

### 1.3 Impact of Scheduled Block Time errors

Assuming an aircraft takes off at the scheduled time it may still arrive earlier or later than scheduled if the Actual Block Time (ABT) differs from the Scheduled Block Time (SBT). Arriving late may cause passenger dissatisfaction, the ground handler may have moved on to another plane, causing the turn around processes to start later and further increase the delay. Being structurally late may even cause the airline to lose its (sometimes very) valuable airport slot, which is the right to use airport facilities at a certain time window during a predefined period. Arriving early will cause both the aircraft and the crew to go idle and will seldom lead to an earlier departure, since passengers will not all be present yet and the original flight plan needs to be adhered to. Scheduling the block times too long may cause less aircraft and crew utilization because less rotations fit the crew schedule. This is easily understood since, all other things being equal, longer SBT's will cause more aircraft arriving early and both crew and aircraft will idly be waiting for the turnaround process. Scheduling block times too short, according to Mayer and Sinai [2] (2003) results in "airlines minimize labor costs at their passengers' expense, although there is also some support in favor of airlines trying to maintain greater aircraft utilization". Scheduling block times too short will also lead to delays, which may propagate throughout the day, which is well described by Beatty et al as early as 1999 [4].

## 2 Part I performance of the current method

### 2.1 Algorithms and design choices

**Performance of the current method** To determine some statistics of the current performance of the scheduled block times, a histogram is made of the block time differences, i.e. the difference between the scheduled block time and the actual block time per leg, resulting in a negative number if the actual block time is shorter than scheduled and a positive number if the actual block time

exceeds the scheduled block time. The percentage of flights with an actual block time shorter than the scheduled block time is calculated to measure current performance. On time performance is measured as the percentage of flights that arrive within 15 minutes of the scheduled arrival time. To clarify the impact of the current method on the arrival on time performance of the company, the percentage of actual block times that exceed the scheduled block time by fifteen minutes or more is also calculated. All other things being the same, that is the percentage of delayed flight inherent to the schedule.

**Distribution of the block times** Box plots, histograms and QQ-norm-plots are made for all routes, to investigate the distribution of the block times and its main components. Also the changes of the distributions of the block times per route will be examined. Observing how the distributions develop over time will help understand why the different models perform the way they do.

## 2.2 Distribution of the block times

The following two histograms show what all histograms display:

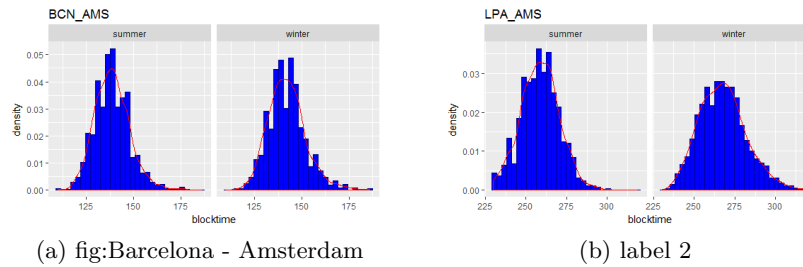


Fig. 1: Distributions resemble normality

The shape of the histograms of the block times shows the typical symmetrical bell shape of the normal distribution.

Applying the Shapiro test for normality to the block times, at 95% confidence level for 25% of the block times normality is not rejected by the test.

The box plots almost all show similar images as below. The median in summer is some of the cases higher in others lower and there are more outliers on the higher percentiles than on the lower percentiles.

The QQ-plots also all are almost perfect straight lines except for the tails.

## 2.3 Performance of the current method to schedule block times

Over the whole period of the dataset, 77.7% of all block times are shorter than or equal to the scheduled block time. Of the block time differences 97.7% is

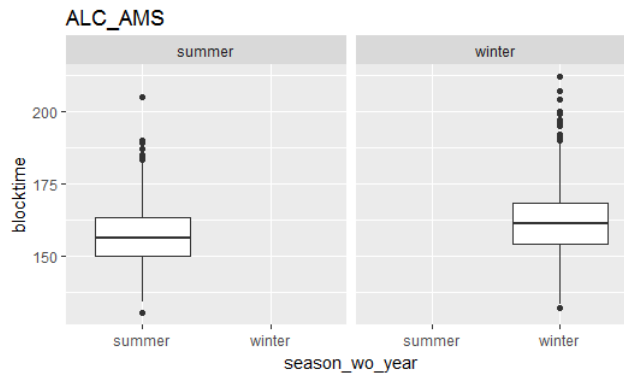


Fig. 2: Typical boxplot 2014 - 2017

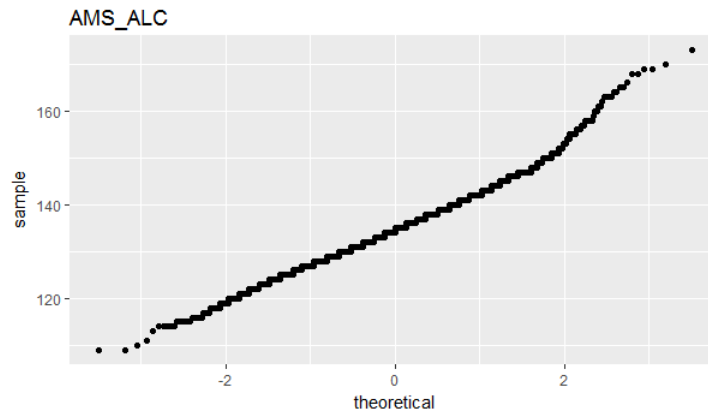


Fig. 3: Typical QQ-plot 2014 - 2017

smaller than 15 minutes. Ergo 2.3% of the arrival delays are inherent to the way block times are planned. If the planning unit would have used exactly the 80% quantile of the block times of summer 2015, mathematically rounded to the nearest 5 minutes to determine the scheduled block times of summer 2016, 80.8% of all actual block times would have been shorter than or equal to the SBT and 98.2% of all legs would have had an ABT less than 15 minutes longer than the SBT.

#### 2.4 Changes in distribution over the years per route

A one way anova performed on the top 30 summer and top 30 winter routes, shows that the means per route do change from year to year. For the 30 routes 16 in summer and 12 in winter the null-hypothesis of the mean being the same over the three years is rejected, with  $\alpha = 0.05$ . However when we look closer at how

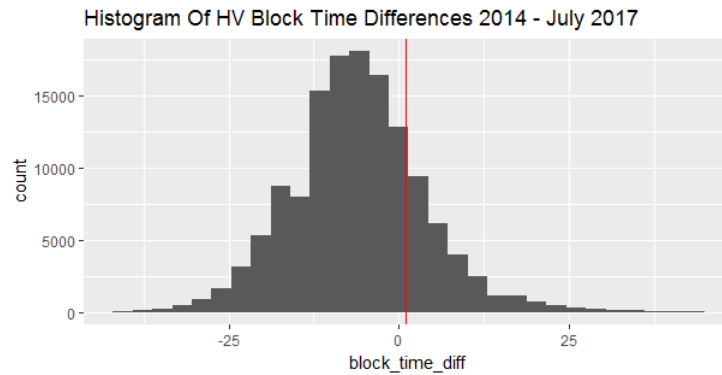


Fig. 4: Histogram of difference between schedule and actual block time

many of the 100 most flown routes have a difference in mean change of at least five minutes between the minimum and the maximum of the three years, there are only four: "AYT\_AMS", "CMN\_AMS", "GZP\_AMS" and "HRG\_AMS", which are all routes longer than 2,300 kilometers, the longer distances. There are only 3 routes on which the difference between the maximum and minimum standard deviations are 3 minutes or more - no standard deviations differ 5 minutes or more over the years Next interesting result is how the standard deviations compare to the means of the different routes: they increase linearly and little with the increase of the block time.

This can also be seen if we plot the actual block times on the distances of the

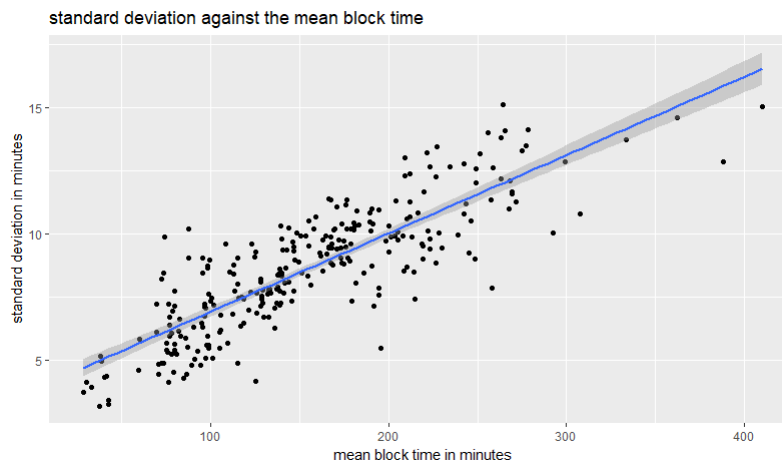


Fig. 5: Standard deviations plotted to mean block times with trend

destinations: the variation of the block times increases as distances increase. The

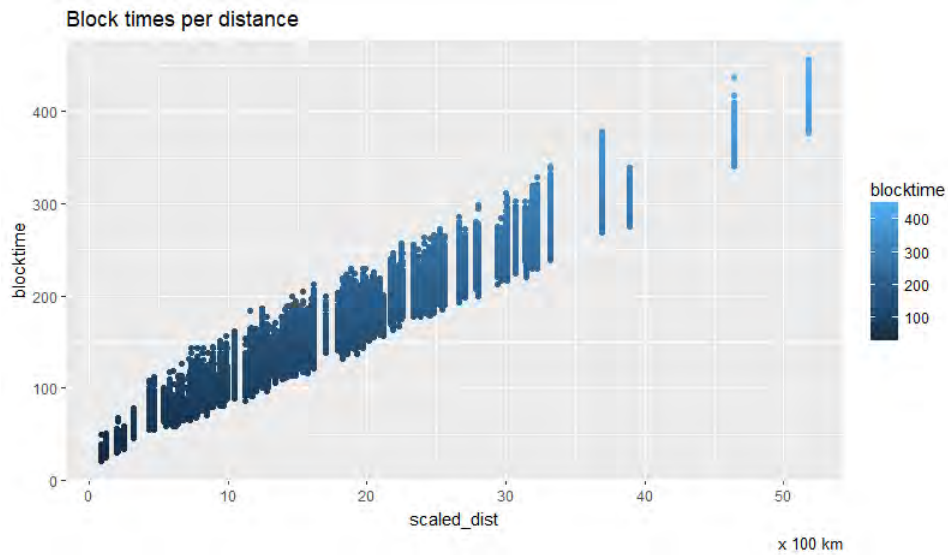


Fig. 6: Distances and block times of Transavia

largest standard deviation is little over 15 for a mean block time of close to 300.

## 2.5 Robustness of the 80% quantile for Transavia

The distributions of the block times all resemble normal distributions. Visually they are all nicely bell shaped around a mean with a slightly longer right tail than left. The longer right tail is easily explained by the fact that the block times are the sum of the taxi-out, flight and taxi-in times. Both taxi-times have a lower bound of zero, but no upper bound and the flight times also have a lower bound (straight line, maximum speed) but only the fuel on board as an upper bound. This tail causes the median to be a little better at predicting next season's block time than the mean.

The distributions are approximately normal, it was shown that the means shift only a little from year to year for the same season and the standard deviations are small. Because of the normality and the small changes in mean and standard deviation over the year, a one minute change in mean or a one minute change in standard deviation causes the 80% quantile to change one minute in the same direction. In table ?? an example is shown of a normal distribution with a mean comparable to the shortest of routes flown by Transavia. One minute mean or standard deviation change, only changes the 80% quantile one minute.

If the distribution would be more skewed the change in mean will start shifting the 80% quantile more.

Table 1: 80% Quantile changes

	sigma										
mu	15	16	17	18	19	20	21	22	23	24	25
<b>82</b>	95	96	96	97	98	99	100	101	101	102	103
<b>83</b>	96	96	97	98	99	100	101	101	102	103	104
<b>84</b>	97	97	98	99	100	101	102	103	103	104	105
<b>85</b>	98	99	99	100	101	102	103	103	104	105	106
<b>86</b>	99	99	100	101	102	103	104	105	105	106	107
<b>87</b>	100	101	101	102	103	104	105	105	106	107	108
<b>88</b>	101	101	102	103	104	105	106	106	107	108	109
<b>89</b>	102	103	103	104	105	106	107	108	108	109	110
<b>90</b>	103	103	104	105	106	107	108	109	109	110	111
<b>91</b>	104	105	105	106	107	108	109	109	110	111	112
<b>92</b>	105	105	106	107	108	109	110	110	111	112	113

## 2.6 Conclusion part I

The way Transavia block times are distributed, symmetrical and small standard deviation, and the way these behave over time, very small changes, makes the 80% percentile over previous similar seasons a very robust method of predicting block times.

## 2.7 Future research

It could be interesting to explore more distributions to find the limits where a 80% or other quantiles are a good rule of thumb that allows 80% of next season's events occur on time.



### 3 Part II Improving on the current method

#### 4 Algorithms and design choices

##### 4.1 Data

For all operations R software and packages will be used. The choice for R is partly personal preference, R can perform all tasks required and the data set is not that large that performance issues might be expected.

The data available for this research is all Transavia flight data from 1-1-2014 until 10-07-2017, which includes scheduled departure and arrival date times, actual off-blocks, take off, touch down and on blocks date times, (IATA) delay reason and duration, aircraft type (Boeing 737-700 or 737-800 mainly). Full list of the data available is added as appendix A. Also public data containing the longitude and latitude of European Airports was downloaded and used to among others calculate distances between Origin / Destination pairs.

First step is to clean up the data, removing features and records of no interest as well as data errors and outliers. There is some iteration between the data cleaning step and the analysis step, since new outliers and impossibilities may become apparent only after deeper inspection.

From the literature and from functionaries of the company, the following relations are assumed to exist and are investigated:

- seasonality;
- difference per day of the week;
- differences per part of the day;
- differences per aircraft type
- differences depending on how much traffic is present.

Visual inspection is performed on the dataset to investigate the distribution of the block times, the presence of these relations and patterns in the block times, over the data as a whole as well as per route.

Statistical tests, specifically the sign-test and one way anova is performed. Clear and present danger of performing these tests repeatedly, is making type I errors. In the evaluation of the results this will be taken in to account and no strong conclusions will be based solely on the outcome of the tests. The tests are used as a quick and easy procedure to check the presence of something worth investigating deeper.

Significance will be determined at the 95% confidence level. There is no point in weakening any conclusions by exchanging the type I errors for type II errors.

The method of Litvay (2012) [3] will be applied as closely as possible to the Transavia data set.

A linear regression and XGboost model are used to attempt to improve on the block time estimates.

The model that initially performs best, will be trained and tested on different cross sections of the data and features.

The performance of all methods will be compared to the median of the train set per OD-pair as a prediction for the test set. This is the simplest of all models. The mean might even be simpler but will only perform better than the median if the distributions are exactly normal, which they are not exactly.

Error measure for each test will be the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) so all errors will be in minutes.

## 5 Experimental set up

### 5.1 Data clean up

A summary of the full original dataset is added in Appendix A. First non-existing and irrelevant flights, i.e. flights without an aircraft registration to it, canceled flights, flights without fuel use, Origin and destination are identical, test and training flights are removed (flight numbers smaller than 71) as well as some other data errors. Details are in the code. These actions reduce the data set to 135.111 records.

### 5.2 Feature creation

Sixty new features are created. A full list is in the Appendix B. Here only the most used and most important features are described: First, all important durations are calculated by subtracting date-times, for instance the actual taxi-in, flight, taxi-out and block times as well as the scheduled block time. Also different combinations of the date time of each leg are created, like the month and week number, the season, day of the week, different subdivisions of the day, for example departure and arrival slots of 20 minutes, hours, 6 hour blocks and the 10/10/4 hours subdivision Litvay uses. The Origin / Destination pair is added (route). From the Open Flights website <https://openflights.org/data.html> the longitude and latitude of all airports was downloaded and join on to the data, which was used to calculate the distance between OD-pairs.

**Visual inspection of trend and seasonality** If there is a strong relation between time of the year and the length of the block time this should become visible. Time series are plotted and inspected for the top 20 most flown routes.

### 5.3 Tests and models

Since, as was shown in the part I, the assumption of normality does not hold for all routes, a test without assumptions with regard to the distribution is needed. To investigate whether block times show different values over different time periods, the sign test was used. Main idea of using this simple statistical test was to have a quick insight in patterns that might be visible over different time periods. The sign test is performed

- block time per route per departure hour;
- block time per route per Litvay-block;
- block time per route per season per Litvay-block;
- block time per route per weekday per Litvay-block;
- block time per route busy yes / no;
- block time per route per aircraft type;

The test is performed for every route. For speedy insight only the p-values below the  $\alpha = 5\%$  threshold are aggregated. Also for evaluating the practical implications of the findings, the results are filter for median differences of 5 minutes or more.

#### 5.4 Advanced models

A linear model and extreme gradient boost, which is a tree based model, are used, to test whether more advanced models can predict block times better. For both algorithms 10 fold cross validation is used to determine parameter settings. For the xgb expandgrid, trainControl and train functions from the R caret package will be used to find optimal parameters. First both models will be run with summers 2014 and 2015 will be used as train set, using summer 2016 as test set and the models will be trained and tested for each route individually - only for routes present in both the train and the test set. Features will be month\_number, weekdaynr and deelblok (the 10/10/4 subdivision).

**Over all routes** Next both models will be trained on all routes, in stead of on each route separately, which opens the possibility to add features that are stationary per individual route, like scaled\_distance and direction and may reveal patterns that are not visible for individual routes.

The best performing model will be trained on the winter of 2014 and 2015 and tested on winter 2016. Finally the calendar years 2014 and 2015 will be used as train set and 2016 as test set. The performance of the models for each test will be compared to the performance of current method of determining the Scheduled Block Times, the train median and the mean per month, per weekday, per part of the day Litvay proposed.

#### 5.5 Dependent variable

With the models the actual block time and not the 80% quantile of the block times is the dependent variable which is predicted.

## 6 Results

### 6.1 Seasonality

Time series were plotted for the top thirty routes. Two plots are displayed here: one that best displays seasonality of all plots inspected, and one that shows what most plots show: no discernible seasonality.

The plot of the block times from Amsterdam to Las Palmas Airport, Gran Canaria, shows some seasonality. The block times from Barcelona to Amsterdam

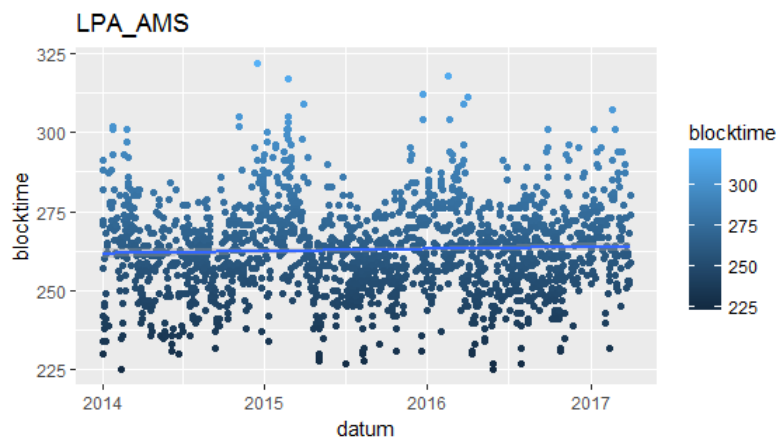


Fig. 7: Actual block times Las Palmas - Amsterdam 2014 - 2017

show a clear trend, but no clear seasonality: in the winter less legs are flown and the shorter block times appear to be less present.

#### **Block time per route per departure hour, weekday, part of the day**

Weekdays show possibly only type *I* errors. Other tests show few significant results - and must contain some type *I* errors. Some indication that morning and day departures result in longer block times, evening departure shorter.

#### **Block time per route busy yes / no**

Busy is defined as the periods in which the number of legs flown by Transavia was greater than the mean of the whole period. Not busy is defined as less legs than average. Here there are 115 significant differences to the median of each route on 498 records. This could be considered a pattern. In figure 9 these 115 significant results are summarized again and during busy periods the block times appear to be more often shorter than the median of the route and during less busy periods more often longer.

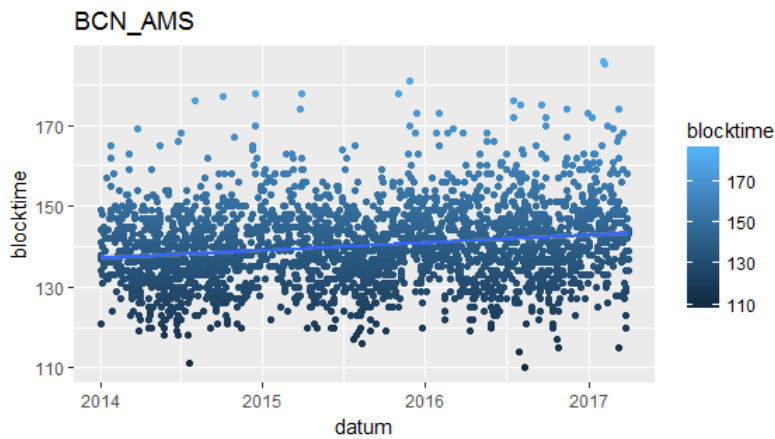


Fig. 8: Actual block times Barcelona Amsterdam 2014 - 2017

busy	korter	all	pval	plus5mins	
no	15	61	0.00	1	0
yes	40	54	0.00	-1	1

Fig. 9: Pattern of busy or not

**Per aircraft type** The subdivision in 737-700's and 737-800's showed clearly, with significant p-values, a difference. After subdividing the data in distances: the 800's are deployed on the longer distance, hence have significantly more often longer block times. After correcting for this the p-values stop being at a significant level.

## 7 Advanced models

### 7.1 Linear model

When we fit a linear model with the "day of the week", "part of the day" and the "month number" as variables to predict the block time, the linear model does better than the median in sixty out of the 150 routes. The average error over all 150 routes is displayed in table 2. The median of the difference between

Table 2: Average error of current method, linear model and Litvay

RMSE.cpu	RMSE.model	RMSE.litv	MAE.cpu	MAE.model	MAE.litv
9,05	9,21	9,80	7,09	7,27	7,74

the MAE of the linear model and the MAE of the median is  $-0.10$  minute.

**XGBoost** Averaged over all 150 routes that are both in the train and in the 2016 test set, table 3 shows RMSE.cpu of the current method, the 80% quantile of last season, the RMSE of the xgboost model and the RMSE for the Litvay-way and on the right the same for the Mean Absolute Error.

Table 3: Average error of current method, XGBoost model and Litvay

RMSE.cpu	RMSE.model	RMSE.litv	MAE.cpu	MAE.model	MAE.litv
9,05	9,75	9,80	7,09	7,69	7,74

When we add two features, "drukte" and "number of passengers", the performance of the xgboost model actually deteriorates to 10.33 averaged over all 150 routes.

There are ten routes on which the model (or the Litvay-way) perform better, so in 140 of the 150 routes using the median of previous periods results in the smallest RMSE.

Expanding the features of the linear model to the following:

- AIRCRAFT\_CAPACITY,
- NUMBER\_OF\_ADULTS,
- NUMBER\_OF\_PAX,
- weeknr,
- YEAR,
- weekdaynr\_x,
- weekdaynr\_y,
- weekend,
- deelblok,
- month\_nbr

increases the error of the linear model: the average RMSE increases to 9.74 and the average MAE to 7.73.

**On the whole set of routes** Training the linear model on all routes, and features summed up in table 4, returns the following feature importance after 10 fold cross validation: The average RMSE of the linear model increases to 10.95. The average RMSE of the train median per route remains 9.05, of course. The average MAE increases to 8.91 and the MAE of the median remains at 7.09.

**XGB on al routes** The average RMSE of the XGBoost model on the same set of variables is 9.51, the average MAE 7.51

Table 4: Feature importance Linear model - all routes

distance	100
aantal_route	3.3689
deelblok	0.5315
YEAR	0.4948
weekend	0.4361
weekdaynr_y	0.3338
weekdaynr_x	0.2909
NUMBER_OF_PAX	0.2886
AIRCRAFT_CAPACITY	0.2481
NUMBER_OF_ADULTS	0.201
weeknr	0.1067
month_nbr	0

**Linear model on winter** The average RMSE of linear model trained on the winter seasons of 2014/2015 and 2015/2016 and tested on the winter of 2016/2017 is 10.70, while the average RMSE of the median is 10.18. The average MAE of the median is: 7.87, of the model 8.37. On one route, "MUC\_EIN" the linear model performed very poorly, causing a RMSE of 8 times as high as the median. When deleted the differences become even smaller. Only one route "AMS\_FNC" was the difference in MAE 5.7 minutes. The median of the differences in MAE is zero. On 49 out of 94 matching winter routes the linear model does better than the mean. When the features "month" and "weekday" are removed, the RMSE decreases to 10.20.

**Linear model on Calender years** The average RMSE of the model is 9.88 compared to the median 9.42. The average MAE of the model is 7.75; of the median: 7.33. The model performs really poor on 4 routes inflating the difference. In 80 out of 172 routes the model does better than the median. The median value of the differences in MAE of all the routes is 0.1 minute.

## 8 Analysis and Discussion

**Applied tests** Applying a statistical test repeatedly creates a serious risk for type I errors, which influences the reliability of the results we see. These can only be seen as a possible indication. Also on some of the tests trying to subdivide the data, leaves them spread over too many bins. The very weak indications that come from the tests are in line with what the seasonality and models show: the patters, if there at all are very weak.

**Seasonality and other patterns in time** The visual inspection of the seasonality of the block times revealed hardly visible seasonality, contrary to expectations of myself and experienced functionaries of Transavia. The lack of clear patterns can be explained by a number of facts.

**Lack of volume** per route per day. Transavia has less than 50 aircraft (depending on the season) in use. The number of data in each subdivision just is not as large as, for example, the data set of SWA that Litvay used. SWA had at the time well over 500 aircraft in use.

**Air Traffic Management Slots** EuroControl <http://www.eurocontrol.int/news/what-slot> or what the FAA calls GDP, Ground Delay Program may well be the best explanation for the lack of seasonality and other patterns: there is a process in place that, for environmental and safety reasons, prevents as much as possible that delays occur once the aircraft is underway.

**IATA Delay Code 89** When the runway on the departing airport is congested, the delay does not show up as a longer taxi-out time, but as a code 89 [/urlhttps://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda-reports/standard-iata-delay-codes-ahm730.pdf](https://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda-reports/standard-iata-delay-codes-ahm730.pdf) : the plane is not allowed to push-back but has to wait at the gate, thus saving fuel and not increasing further the already congested taxi routes to the runway.

**IATA Delay Code 81** If any of the sectors en route, i.e. for the exact route for which a flight plan was filed, is congested either by too many aircraft wanting to cross the same airspace at the same altitude at the same time so that minimum distances can not be maintained, or if there are just not enough Air Traffic Controllers to safely handle all passing aircraft, the aircraft is not allowed to push back. Instead it suffers a departure delay at the gate, IATA delay code 81.

**IATA Delay Code 83** It is also possible for the destination airport to be congested, which also causes a departure delay if the aircraft is not already on its way, code 83 ATFM due to RESTRICTION AT DESTINATION AIRPORT.

**Code 81 avoidance** Code 81 is the second most frequent delay suffered by Transavia (I do not think Transavia will appreciate me summing up actual figures here). These en-route delays can be avoided if the Operational Controller of Transavia files a new flight plan, circumnavigating the congested area. Obviously they will only do that if the time it takes to execute the new flight plan is shorter than the original flight plan plus the expected wait time for the ATM-slot.

**Overall** the seasonality of the congestion and weather is definitely there. Most of it is managed away and never reaches the block times, but is captured by the process in departure delays. This system of slots, causes the actual block times to have little variation and seasonality. The variation in the block times that is left is caused by other causes that occur once the plane has pushed back. These can be weather or congestion related, but become apparent only after the plane is en route. The variation may also be caused by pilots asking for, and being granted, a direct route, but most importantly it is the actual route flown.



**Models over all routes in stead of per individual route** The poor performance of the models over all routes, is likely to be caused that patterns are not the same over all routes. The XGBoost model did better on the overall routes, where the linear model does better per route. Possibly if more years of data is available results of the XGBoost model might improve some more.

**Future research** It would be a challenge to be able to completely explain the variation of block times in hindsight, using the exact route flown and wind conditions en route. These data are not yet available, but when that happens it would be clear exactly what forces make block times vary, and that might make it possible to either predict or control the variance better.

## 9 Conclusions

Conclusion is that there are few obvious patterns in the block times. Those that may be visible, are dependent on routes: some routes display the patterns others do not, or show opposite patterns. The differences are too small to be of much business relevance, but may be used to apply small tweaks to improve on the schedule. Linear regression and a more advanced machine learning method can not capture the patterns that may be present to predict block times better than the median, making the median the best predictor for next season's block times and the 80% quantile a good scheduled block time if the aim is to have 80% of next season's aircraft to arrive on time.

## References

1. Hao, Lu and Hansen, Mark (2013): How Airlines Set Scheduled Block Times. 10th USA/Europe Air Traffic Management Research and Development Seminar, Chicago IL
2. Mayer, C., & Sinai, T. (2003): Why do airlines systematically schedule their flights to arrive late?. The Wharton School, University of Pennsylvania.
3. Litvay, R.O. (2012): Development and Validation of a New Air Carrier Block Time Prediction Model and Methodology. The Ohio State University Provided by the SAO/NASA Astrophysics Data System
4. Beatty, R., Hsu, R., Berry, L., & Rome, J. (1999). Preliminary evaluation of flight delay propagation through an airline schedule. *Air Traffic Control Quarterly*, 7(4), 259-270.
5. Sohoni, M., Lee, Y. C., & Klabjan, D. (2011). Robust airline scheduling under block-time uncertainty. *Transportation Science*, 45(4), 451-464.

## A Appendix Available Data

```
'data.frame': 197416 obs. of 95 variables:
 $ ..AIRPORT_PLANNED : chr "" "" "SXF" "" ...
 $ COUNTRY_PLANNED : chr "" "" "DE" "" ...
 $ AIRPORT_ACTUAL : chr "" "" "SXF" "" ...
 $ COUNTRY_ACTUAL : chr "" "" "DE" "" ...
 $ AIRPORT_CODE : chr "MAD" "MAD" "MUC" "OPO" ...
 $ HANDLING_FOR_AIRLINE : chr "HV" "HV" "HV" "HV" ...
 $ AREA_MANAGER : chr "functionary names removed" ...
 $ AIRCRAFT_REGISTRATION : chr "PHHZW" "PHHZW" "PHXRV" "PH-
HXC" ...
 $ AIRLINE : chr "KL" "KL" "HV" "HV" ...
 $ REPORTING_AIRLINE : chr "HV" "HV" "HV" "HV" ...
 $ LEG_NUMBER : int 1700 1704 9515 6507 9352 5496 5292 6332 9216 6654 ...
 $ SUFFIX : chr "" "" "" "" "" ...
 $ LEGDATE : chr "2017-03-29 08:20:00" "2017-03-29 15:10:00" ...
 $ CANCELLED : chr "" "" "" "" "" ...
 $ AIRPORT_FROM_IATA_ : chr "MAD" "MAD" "MUC" "OPO" ...
 $ AIRPORT_TO_IATA_ : chr "AMS" "AMS" "SXF" "FUE" ...
 $ SCHEDULED_AIRPORT_TO_IATA_ : chr "AMS" "AMS" "SXF" "FUE" ...
 $ COMPOSITION_FLIGHTDECK : chr "" "" "C1F1" "" ...
 $ COMPOSITION_CABIN : chr "" "" "1P 2CA1" "" ...
 $ ON_TIME_Y_N_ : int 0 0 1 0 1 0 1 1 1 0 ...
 $ FLIGHTTYPE_CODE : chr "S" "S" "J" "J" ...
 $ HANDLING_AGENT_ON_BLOCKS : chr "2017-03-29 11:10:00" "2017-03-29
17:51:00" ...
 $ HANDLING_AGENT_TOUCHDOWN : chr "2017-03-29 10:57:00" "2017-03-
29 17:38:00" ...
 $ HANDLING_AGENT_AIRBORNE : chr "2017-03-29 08:52:00" "2017-03-29
15:37:00" ...
 $ HANDLING_AGENT_OFF_BLOCKS : chr "2017-03-29 08:38:00" "2017-03-29
15:26:00" ...
 $ ACTUAL_ON_BLOCKS : chr "2017-03-29 11:10:00" "2017-03-29 17:51:00" ...
 $ ACTUAL_TOUCHDOWN : chr "2017-03-29 10:57:00" "2017-03-29 17:38:00"
...
 $ ACTUAL_AIRBORNE : chr "2017-03-29 08:52:00" "2017-03-29 15:37:00" ...
 $ ACTUAL_OFF_BLOCKS : chr "2017-03-29 08:38:00" "2017-03-29 15:26:00" ...
 $ AIRCRAFT_ON_BLOCKS : chr "" "" "2017-03-28 17:45:00" "2017-03-28 10:42:00"
...
 $ AIRCRAFT_TOUCHDOWN : chr "" "" "2017-03-28 17:39:00" "2017-03-28
10:36:00" ...
 $ AIRCRAFT_AIRBORNE : chr "" "" "2017-03-28 16:49:00" "2017-03-28 08:30:00"
...
 $ AIRCRAFT_OFF_BLOCKS : chr "" "" "2017-03-28 16:35:00" "2017-03-28
08:11:00" ...
```

\$ HEAVY\_CABIN\_CREW\_Y\_N\_ : chr "" "" "N" "" ...  
 \$ HEAVY\_FLIGHTDECK\_CREW\_Y\_N\_ : chr "" "" "N" "" ...  
 \$ SERVICE\_TYPE : chr "J" "J" "J" "J" ...  
 \$ AIRPORT\_NAME\_TO : chr "AMSTERDAM - SCHIPHOL INTERNATIONAL"  
 "AMSTERDAM - SCHIPHOL INTERNATIONAL" "BERLIN SCHONEFELD  
 APT" "PUERTO DEL ROSARIO" ...  
 \$ AIRPORT\_CITY\_TO : chr "AMSTERDAM" "AMSTERDAM" "BERLIN"  
 "FUERTEVENTURA" ...  
 \$ AIRPORT\_COUNTRY\_CODE\_TO : chr "NL" "NL" "DE" "ES" ...  
 \$ AIRPORT\_NAME\_FROM : chr "MADRID BARAJAS APT" "MADRID BARA-  
 JAS APT" "FRANZ JOSEF STRAUSS" "PORTO" ...  
 \$ AIRPORT\_CITY\_FROM : chr "MADRID" "MADRID" "MUNICH" "PORTO"  
 ...  
 \$ AIRPORT\_COUNTRY\_FROM : chr "ES" "ES" "DE" "PT" ...  
 \$ AIRPORT\_NAME\_TOP : chr "AMSTERDAM - SCHIPHOL INTERNATIONAL"  
 "AMSTERDAM - SCHIPHOL INTERNATIONAL" "BERLIN SCHONEFELD  
 APT" "PUERTO DEL ROSARIO" ...  
 \$ AIRPORT\_CITY\_ARRIVAL\_TOP : chr "AMSTERDAM" "AMSTERDAM"  
 "BERLIN" "FUERTEVENTURA" ...  
 \$ AIRPORT\_COUNTRY\_CODE\_TOP : chr "NL" "NL" "DE" "ES" ...  
 \$ AIRLINE\_CODE : chr "KL" "KL" "HV" "HV" ...  
 \$ AIRLINE\_DESCRIPTION : chr "KLM ROYAL DUTCH AIRLINES" "KLM  
 ROYAL DUTCH AIRLINES" "TRANSAVIA AIRLINES" "TRANSAVIA AIR-  
 LINES" ...  
 \$ ICAO\_AIRLINE\_DESIGNATOR : chr "KLM" "KLM" "TRA" "TRA" ...  
 \$ AIRCRAFT\_TYPE : chr "73H" "73H" "73W" "73H" ...  
 \$ AIRCRAFT\_NAME : chr "Boeing 737-800 Winglets" "Boeing 737-800 Winglets"  
 "Boeing 737-700 Winglets" "Boeing 737-800 Winglets" ...  
 \$ AIRCRAFT\_CAPACITY : int 189 189 149 189 149 189 149 189 149 189 ...  
 \$ AIRCRAFT\_GENERAL\_TYPE : chr "737" "737" "737" "737" ...  
 \$ AIRCRAFT\_VERSION : chr "738H" "738H" "73W" "738H189" ...  
 \$ AIRCRAFT\_CATEGORY : chr "J" "J" "J" "J" ...  
 \$ AIRCRAFT\_INDICATOR\_WHITE\_BODY : chr "N" "N" "N" "N" ...  
 \$ DEPARTURE\_DELAY\_MIN\_ : int 18 16 5 31 0 35 0 6 0 20 ...  
 \$ NUMBER\_OF\_FLIGHTS\_STD\_MINUS\_0 : int 0 0 0 0 0 0 0 0 0 0 ...  
 \$ NUMBER\_OF\_FLIGHTS\_STD\_0 : int 0 0 0 0 1 0 1 0 1 0 ...  
 \$ NUMBER\_OF\_FLIGHTS\_STD\_15 : int 0 0 1 0 0 0 0 1 0 0 ...  
 \$ NUMBER\_OF\_FLIGHTS\_STD\_60 : int 1 1 0 1 0 1 0 0 0 1 ...  
 \$ NUMBER\_OF\_FLIGHTS\_STD\_180 : int 0 0 0 0 0 0 0 0 0 0 ...  
 \$ NUMBER\_OF\_FLIGHTS\_STD\_PLUS\_180 : int 0 0 0 0 0 0 0 0 0 0 ...  
 \$ NUMBER\_OF\_ADULTS : int NA NA 110 160 127 157 58 181 105 150 ...  
 \$ NUMBER\_OF\_CHILDREN : int NA NA 1 5 2 0 2 0 3 1 ...  
 \$ NUMBER\_OF\_INFANTS : int NA NA 0 3 0 0 0 0 1 1 ...  
 \$ NUMBER\_OF\_DISEMBARKING : int NA NA 0 0 0 0 0 0 0 0 ...  
 \$ NUMBER\_OF\_EMBARKING : int NA NA 111 0 129 157 60 181 109 152 ...

\$ NUMBER\_OF\_TRANSIT : int NA NA NA NA NA NA NA NA NA NA ...  
\$ NUMBER\_OF\_PASSENGERS : int 0 0 111 165 129 157 60 181 108 151 ...  
\$ NUMBER\_OF\_PAX : int 0 0 111 168 129 157 60 181 109 152 ...  
\$ FUEL\_USED\_IN\_LITERS : int 0 0 2627 6947 10610 5431 4332 7010 5848  
8084 ...  
\$ FUEL\_USED\_IN\_GALLONS : int 0 0 694 1835 2803 1435 1144 1852 1545  
2136 ...  
\$ FUEL\_UPLIFT : chr "" "" "4350" "1750" ...  
\$ SCHEDULE\_DEPARTURE : chr "2017-03-29 08:20:00" "2017-03-29 15:10:00"  
"2017-03-28 16:30:00" "2017-03-28 07:40:00" ...  
\$ SCHEDULE\_ARRIVAL : chr "2017-03-29 10:55:00" "2017-03-29 17:50:00"  
"2017-03-28 17:40:00" "2017-03-28 10:00:00" ...  
\$ NUMBER\_OF\_LEGS : int 1 1 1 1 1 1 1 1 1 1 ...  
\$ DEPARTURE\_DELAY\_INDICATOR : chr "Y" "Y" "Y" "Y" ...  
\$ MK\_DIM\_LEG : int 17546391 17546392 17546404 17546409 17547169 17547170  
17547172 17547173 17547174 17547031 ...  
\$ MK\_DIM\_OPERATING\_AIRLINE : int 138 138 113 113 113 113 113 113 113  
113 ...  
\$ MK\_DIM\_AIRPORT\_DEPARTURE : int 2743 2743 3071 3445 4605 4877 4910  
4922 4922 85 ...  
\$ MK\_DIM\_AIRPORT\_DEST\_PLANNED : int 169 169 4403 1448 3071 169 3996  
169 3071 1227 ...  
\$ MK\_DIM\_AIRPORT\_DEST\_ACTUAL : int 169 169 4403 1448 3071 169 3996  
169 3071 1227 ...  
\$ MK\_DIM\_AIRPORT\_DIVERSION : int -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
\$ MK\_DIM\_AIRCRAFT : int 156 156 181 32547 181 32545 185 52545 178 32549  
...  
\$ MK\_DIM\_HANDLING\_AGENT : int 0 0 3061 34 28 9 3069 49 49 9 ...  
\$ MK\_DIM\_HANDLING\_SUPERVISOR : int 0 0 0 0 0 0 0 0 0 ...  
\$ MK\_DIM\_COMMERCIAL\_ROUTE : int 3354 3354 77207 0 79225 300 363 47  
66321 321 ...  
\$ FLIGHT\_ID : chr "KL169920170329AMSMAD" "KL170320170329AMSMAD"  
"HV951520170328MUCSXF" "HV650720170328AMSOPO" ...  
\$ YEAR\_WEEK : int 201713 201713 201713 201713 201713 201713 201713  
201713 201713 201713 ...  
\$ YEARMONTH : int 201703 201703 201703 201703 201703 201703 201703  
201703 201703 201703 ...  
\$ DELAY : int NA NA NA 20 NA 35 NA 6 NA 20 ...  
\$ SEQUENCENO : int NA NA NA 1 NA 1 NA 1 NA 1 ...  
\$ IND\_CORRECTED : chr "" "" "" "N" ...  
\$ REASONCODE : int NA NA NA 93 NA 55 NA 89 NA 93 ...  
\$ REASONALTCODE : logi NA NA NA NA NA NA NA ...

## **B Appendix created features**

## **C Appendix full result of linear model**

rankdef == TRUE indicates that some covariates of the linear model are collinear.  
Once on LPA\_AMS this leads to a ridiculous model-result.

## **D Appendix full result of XGBoost model**

```

$ weeknr|      : num  13 13 13 13 13 13 13 9 8 9 ...
$ season      : chr  "winter1617" "winter1617" "winter1617" "winter1617" ...
$ season_wo_year : chr  "winter" "winter" "winter" "winter" ...
$ YEAR       : num  2017 2017 2017 2017 2017 ...
$ season_wo_year_numeric : num  0 0 0 0 0 0 0 0 0 ...
$ route      : chr  "MUC_SXF" "TLV_MUC" "VCE_AMS" "VIE_RTM" ...
$ weekday    : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tues"<...: 3 4 4 4 4 4 3 4 4 5 ...
$ weekdaynr  : num  3 4 4 4 4 4 3 4 4 5 ...
$ weekdaynr_x : num  0.707 0.5 0.5 0.5 0.5 ...
$ weekdaynr_y : num  0.707 0.866 0.866 0.866 0.866 ...
$ weekend     : num  0 0 0 0 0 0 0 0 0 ...
$ taxi_out   : num  14 15 9 17 9 11 14 6 8 15 ...
$ taxi_in    : num  6 4 9 3 19 4 5 4 3 4 ...
$ flight_time : num  50 218 102 93 132 126 154 257 104 72 ...
$ blocktime  : num  70 237 120 113 160 141 173 267 115 91 ...
$ sched_blocktime : num  70 260 120 110 155 145 180 275 120 85 ...
$ datum      : Date, format: "2017-03-28" "2017-03-29" "2017-03-29" ...
$ dep_timestring : chr  "16:30:00" "09:25:00" "11:30:00" "07:20:00" ...
$ rank       : int  2 1 1 1 1 1 1 1 1 1 ...
$ weeknr_x   : num  0.0296 0.0296 0.0296 0.0296 0.0296 ...
$ weeknr_y   : num  1 1 1 1 1 ...
$ block_time_diff : num  0 -23 0 3 5 -4 -7 -8 -5 6 ...
$ toHB       : num  0 1 1 1 1 1 1 0 0 1 ...
$ aantal_route : int  303 85 96 181 215 59 151 51 129 199 ...
$ lat_from   : num  48.4 32 45.5 48.1 39.5 ...
$ long_from  : num  11.786 34.887 12.352 16.57 -0.482 ...
$ lat_to     : num  52.4 48.4 52.3 52 52.3 ...
$ long_to    : num  13.52 11.79 4.76 4.44 4.76 ...
$ distance   : num  464808 2657986 937959 965828 1482773 ...
$ scaled_dist : num  4.65 26.58 9.38 9.66 14.83 ...
$ step_dist  : num  1 2 1 1 1 1 2 3 1 1 ...
$ month_nbr  : num  3 3 3 3 3 3 3 2 3 ...
$ schedule_departure_time : POSIXct, format: "2017-08-04 16:30:00" "2017-08-04 09:25:00" "2017-08-04 17:40:00" ...
$ SDD_slot   : num  50 29 35 22 24 24 54 33 31 25 ...
$ SDD_hours  : num  17 10 12 8 8 8 18 11 11 9 ...
$ SDD_hours_x : num  -0.259 -0.866 -1 -0.5 -0.5 ...
$ SDD_hours_y : num  -9.66e-01 5.00e-01 1.22e-16 8.66e-01 8.66e-01 ...
$ SDD_4bloks : num  3 2 2 2 2 2 3 2 2 2 ...
$ SDD_4bloks_x : num  0.707 0.866 0.866 0.866 0.866 ...
$ SDD_4bloks_y : num  0.707 0.5 0.5 0.5 0.5 ...
$ schedule_arrival_time : POSIXct, format: "2017-08-04 17:40:00" "2017-08-04 13:45:00" "2017-08-04 17:40:00" ...
$ SAD_slot   : num  53 42 41 28 32 31 63 46 37 29 ...
$ SAD_hours  : num  18 14 14 10 11 11 21 16 13 10 ...
$ SAD_hours_x : num  -1.84e-16 -8.66e-01 -8.66e-01 -8.66e-01 -9.66e-01 ...
$ SAD_hours_y : num  -1 -0.5 -0.5 0.5 0.259 ...
$ SAD_4bloks : num  3 3 3 2 2 2 4 3 3 2 ...
$ SAD_4bloks_x : num  0.707 0.707 0.707 0.866 0.866 ...
$ SAD_4bloks_y : num  0.707 0.707 0.707 0.5 0.5 ...
$ deeblok    : num  2 2 2 1 1 1 2 2 2 2 ...
$ direction  : num  1.16 2.53 2.41 2.83 1.18 ...
$ direction_string : chr  "NE" "NW" "NW" "W" ...
$ tot        : int  98 110 110 110 110 110 98 106 98 115 ...
$ meanlegs   : num  115 115 115 115 115 ...
$ drukte     : num  -1 -1 -1 -1 -1 -1 -1 -1 -1 1 ...
$ TN         : int  34 105 105 105 105 105 105 34 36 73 28 ...
$ median_BT  : num  74 247 115 102 149 137 168 264 115 78 ...
$ BT80       : num  82 256 122 108 159 ...
$ FU0500     : num  2843 10863 5100 4540 6947 ...
$ FU0995     : num  4112 14573 6886 6271 9170 ...
$ FU0005     : num  2369 9242 4065 3772 1333 ...
$ strange_fuel : num  0 0 0 0 0 0 0 0 0 ...

```

Fig. 10: Full set of features

Table 5: Results of linear model

route	RMSE.cpu	RMSE.model	RMSE.litv	MAE.cpu	MAE.model	MAE.litv	rankdef
AMS_KGS	8.7	8.7	9.04	7	7.1	7.3	FALSE
KGS_AMS	8.29	8.45	8.88	6.6	6.8	7.2	FALSE
AGP_AMS	11.36	11.29	12.03	8.9	8.9	9.4	FALSE
VIE_RTM	6.46	6	6.71	5.2	4.7	5.4	FALSE
AGP_RTM	9.36	9.43	9.67	7.4	7.5	7.7	FALSE
ALC_EIN	8	8.14	8.15	6.3	6.5	6.8	FALSE
AMS_CHQ	9.37	9.12	9.93	7.7	7.5	8.5	FALSE
AMS_FAO	10.07	10.12	10.83	7.9	8	8.5	FALSE
AMS_NCE	7.43	7.43	8.1	5.6	5.7	6.3	FALSE
AMS_OPO	9.08	8.85	9.57	7.3	7.1	7.7	FALSE
AMS_IBZ	8.67	8.83	9.51	6.9	7	7.5	FALSE
AMS_LCA	8.34	8.9	9.6	6.7	7.5	7.9	FALSE
AMS_RAK	13.17	12.78	13.34	10.7	10.4	10.9	FALSE
LIS_AMS	10.96	10.64	11.48	8.5	8.3	9.3	FALSE
ALC_AMS	10.72	10.54	11.42	8.5	8.4	9	FALSE
AMS_ACE	12.32	12.17	12.43	9.7	9.7	9.9	FALSE
AMS_AGP	10.38	10.49	11.23	8.1	8.2	8.8	FALSE
AGP_EIN	9.57	9.7	10.24	7.6	7.7	8.1	FALSE
AMS_BCN	9.24	9.39	9.81	7.2	7.4	7.6	FALSE
AMS_GRO	9.01	9.14	9.98	6.7	7.1	7.6	FALSE
ALC_RTM	7.61	7.98	9.17	5.8	6.2	7.3	FALSE
TFS_AMS	13.4	13.38	15.4	10.7	10.8	11.5	FALSE
TFS_LPA	6.59	6.19	6.01	5.5	5	4.9	FALSE
TLV_AMS	10.8	11.75	11.97	8.7	9.8	9.8	FALSE
FAO_AMS	11.67	11.4	12.33	9.1	8.9	9.6	FALSE
FAO_EIN	8.97	9.44	9.98	7.2	7.4	8	FALSE
FAO_GRQ	10.55	10.6	11.24	8.2	8.5	9.3	FALSE
HER_AMS	9.37	9.36	10.1	7.4	7.5	8.1	FALSE
IBZ_AMS	9.66	9.58	9.81	7.5	7.5	7.7	FALSE
LIS_EIN	9.49	9.83	11.15	7.1	7.4	8.7	FALSE
PSA_AMS	8.46	8.5	9.03	6.7	6.7	7.3	FALSE
ACE_AMS	11.42	11.53	12.79	9.1	9	10.3	FALSE
CMN_AMS	12.04	12.17	13.9	9.7	9.7	11.1	FALSE
CPH_EIN	5.78	5.77	6.58	4.1	4.1	4.9	FALSE
CTA_AMS	10.25	10.09	10.32	8.2	8.1	8.3	FALSE
AMS_BOJ	10.71	11.23	12.76	8.3	8.1	8.8	FALSE
AMS_BRI	6.96	7.02	7.02	5.5	5.5	5.5	FALSE
AMS_CMN	10.79	10.91	12	8.6	8.8	9.7	FALSE
AMS_ALC	9.85	10.06	10.69	7.5	7.8	8.3	FALSE
AMS_AYT	8.77	9.22	10.06	7	7.3	7.9	FALSE
AMS_LIS	10.82	10.84	11.07	8.3	8.4	8.5	FALSE
AMS_PSA	7.92	8.01	8.78	6.2	6.3	6.9	FALSE
AMS_TLV	9.98	11.03	10.21	7.7	8.5	7.9	FALSE
AMS_RHO	11.04	10.83	10.26	8.3	8.1	7.8	FALSE
AMS_SKG	7.53	7.21	8.08	5.9	5.7	6.4	FALSE
GZP_AMS	7.89	8.02	9.5	6.2	6.3	7.7	FALSE
IBZ_EIN	7.81	7.83	7.92	5.8	5.9	6.1	FALSE
INN_AMS	7.03	7.36	7.11	5.8	6	5.7	FALSE
LPA_RTM	13.14	12.94	12.72	9.7	9.6	9.8	TRUE
AMS_ATH	8.58	8.66	9.24	6.8	6.9	7.4	FALSE
FUE_AMS	13.34	13.46	15.45	10.8	10.9	13.1	FALSE
AGA_AMS	12.36	12.67	13.31	9.5	9.6	10.1	TRUE
AMS_INN	6.22	6.91	7.86	5.2	5.9	6.4	FALSE
AMS_LPA	13.37	13.3	14.49	10.8	10.7	11.4	FALSE
AMS_NAP	7.48	7.55	7.87	5.9	6	6.2	FALSE
AMS_BJV	9.76	9.55	9.1	7.6	7.3	7.1	FALSE
AMS_CTA	8.22	8.34	8.36	6.3	6.4	6.3	FALSE
NAP_AMS	9.34	9.59	10.47	7.2	7.5	8.2	FALSE
BCN_AMS	10.86	10.75	11.27	8.3	8.3	8.7	FALSE

Table 6: result of XGBoost

route	RMSE.cpu	RMSE.model	RMSE.litv	MAE.cpu	MAE.model	MAE.litv
AMS_KGS	8,7	8,9	9,04	7	7,2	7,3
KGS_AMS	8,29	8,53	8,88	6,6	6,9	7,2
AGP_AMS	11,36	11,82	12,03	8,9	9,3	9,4
VIE_RTM	6,46	6,34	6,71	5,2	5	5,4
AGP_RTM	9,36	9,62	9,67	7,4	7,7	7,7
ALC_EIN	8	8,65	8,15	6,3	7,1	6,8
AMS_CHQ	9,37	9,16	9,93	7,7	7,7	8,5
AMS_FAO	10,07	10,67	10,83	7,9	8,4	8,5
AMS_NCE	7,43	7,89	8,1	5,6	6,1	6,3
AMS_OPO	9,08	9,33	9,57	7,3	7,6	7,7
AMS_IBZ	8,67	9,15	9,51	6,9	7,2	7,5
AMS_LCA	8,34	9,49	9,6	6,7	7,8	7,9
AMS_RAK	13,17	13,29	13,34	10,7	10,9	10,9
LIS_AMS	10,96	11,3	11,48	8,5	9,2	9,3
ALC_AMS	10,72	11,32	11,42	8,5	8,9	9
AMS_ACE	12,32	12,75	12,43	9,7	10,1	9,9
AMS_AGP	10,38	11,2	11,23	8,1	8,8	8,8
AGP_EIN	9,57	10,3	10,24	7,6	8,2	8,1
AMS_BCN	9,24	9,75	9,81	7,2	7,6	7,6
AMS_GRO	9,01	10,05	9,98	6,7	7,7	7,6
ALC_RTM	7,61	8,83	9,17	5,8	7,1	7,3
TFS_AMS	13,4	16,81	15,4	10,7	12,9	11,5
TFS_LPA	6,59	6,01	6,01	5,5	4,9	4,9
TLV_AMS	10,8	12,33	11,97	8,7	9,8	9,8
FAO_AMS	11,67	12,06	12,33	9,1	9,4	9,6
FAO_EIN	8,97	10,1	9,98	7,2	8	8
FAO_GRQ	10,55	11,25	11,24	8,2	9,4	9,3
HER_AMS	9,37	10,4	10,1	7,4	8,3	8,1
IBZ_AMS	9,66	10,07	9,81	7,5	7,9	7,7
LIS_EIN	9,49	11,02	11,15	7,1	8,5	8,7
PSA_AMS	8,46	8,85	9,03	6,7	7,1	7,3
ACE_AMS	11,42	12,81	12,79	9,1	10,3	10,3
CMN_AMS	12,04	13,71	13,9	9,7	11,1	11,1
CPH_EIN	5,78	7,35	6,58	4,1	5,6	4,9
CTA_AMS	10,25	10,63	10,32	8,2	8,5	8,3
AMS_BOJ	10,71	12,85	12,76	8,3	9,5	8,8
AMS_BRI	6,96	7,35	7,02	5,5	5,9	5,5
AMS_CMN	10,79	10,92	12	8,6	8,6	9,7
AMS_ALC	9,85	10,58	10,69	7,5	8,2	8,3
AMS_AYT	8,77	9,97	10,06	7	7,8	7,9
AMS_LIS	10,82	11,04	11,07	8,3	8,5	8,5
AMS_PSA	7,92	8,83	8,78	6,2	7	6,9
AMS_TLV	9,98	12,91	10,21	7,7	9,9	7,9
AMS_RHO	11,04	10,22	10,26	8,3	7,8	7,8
AMS_SKG	7,53	8	8,08	5,9	6,4	6,4
GZP_AMS	7,89	9,24	9,5	6,2	7,5	7,7
IBZ_EIN	7,81	8,1	7,92	5,8	6,1	6,1
INN_AMS	7,03	7,48	7,11	5,8	6,1	5,7
LPA_RTM	13,14	12,71	12,72	9,7	9,9	9,8
AMS_ATH	8,58	9,09	9,24	6,8	7,3	7,4
FUE_AMS	13,34	14,99	15,45	10,8	11,9	13,1
AGA_AMS	12,36	13,09	13,31	9,5	9,8	10,1
AMS_INN	6,22	7,82	7,86	5,2	6,4	6,4
AMS_LPA	13,37	14,53	14,49	10,8	11,5	11,4
AMS_NAP	7,48	7,76	7,87	5,9	6,1	6,2
AMS_BJV	9,76	9,28	9,1	7,6	7,3	7,1
AMS_CTA	8,22	8,64	8,36	6,3	6,7	6,3
NAP_AMS	9,34	10,15	10,47	7,2	7,9	8,2
BCN_AMS	10,86	11,06	11,27	8,3	8,6	8,7