Baana predictions

Verna Koskinen

Project in practical machine learning
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, April 11, 2016

Contents

1 Introduction

Baana is a 1.3 km long top-level cycling and pedestrian way in the center of Helsinki, travelling from the Central Railway Station to Ruoholahti. It was opened in 12th of June in 2012 and travels through an old railway deep with two way cycling paths and no traffic junctions. The route has gained a lot of popularity amongst cyclists right after it's opening, most likely due to the safety and velocity of the track. The number of cyclists using Baana each day is calculated using cameras and automatic cyclist detection and is made openly available at Avoin Data service [3] [5].

The purpose of this project is to try and predict future cyclist counts based on weather forecasts. The outcome of the project is a web service where predictions for future cyclist counts are visible as well as past predictions with comparison to the actual value. The web service uses data collected during this project with a precalculated model for the predictions.

1.1 Related work

Baanamittari is a service for displaying info-graph with cyclist counts at Baana together with average temperatures and rain amounts [6]. The service does not provide predictions, but it's easy to see from the graph that there seems to be some correlation between especially the average temperature and cyclist counts. Baanamittari has served as inspiration for starting this project.

To support the notion that weather affects cyclist counts, a research looking into how weather affects commuting traffic within cyclists found that weather can be used to explain about half the variation in commuting cyclist counts in Australia [1]. Especially rainfall and temperature were seen to have effect on the counts.

Another study in Louisiana showed that linear methods for predicting traffic speed are preferable to k-NN and kernel smoothing methods [7]. Traffic speed predictions are similar with cyclist count predictions so similar results may apply here as well.

2 Data

Important part of this project was to collect and use available open source data to implement a machine learning system. The raw data acquired was not in a shape to be directly inserted into the algorithms so it needed to be cleaned and sanitised before usage. This section describes the process of acquiring and cleaning the data.

Cleaned data is stored as flat files, containing weather data, cyclist counts and cyclist predictions on their own files with each day on its own line.

2.1 Cyclist counts

The data for the cyclist counts is provided by Helsingin kaupunkisuunnitteluvirasto and is openly available and updated around half yearly at Avoin Data service [3]. The data used for this project spans over about three years, from 1st January 2013 until today. Data within that time span but not yet available from Avoin Data was helpfully provided by Helsingin Kaupunkisuunnitteluvirasto over email. Some of the data is in Excel files and some in CSV format, so two different data readers needed to be implemented.

The time span consist of about 1200 daily counts. Hourly counts would also have been available, but daily counts for the predictions was chosen as it seems more meaningful to predict. Some of the raw data acquired for the project was in hourly count format but was transformed into daily counts during the data cleaning phase.

Data after end of January is taken from Infracontrol real time display [4] daily at the end of the day, because the data by Kaupunkisuunnitteluvirasto is updated only half yearly.

2.2 Weather information

Weather information for the same time span with the cyclist counts was acquired from the Finnish Meteorological Institutions open data service [2]. Daily weather data containing rain amounts, average temperature, days high- and low temperatures and the depth of snow is used for this project. The data is collected from the FMI open data service in a very old fashioned XML format, but is cleaned and saved for the machine learning algorithm to

use, as a two dimensional array containing each days values on their own line.

The raw data contains either -1 or no value for snow when there is none. All empty values have been turned to -1. 0 snow most likely means there is some snow, but less than 1 cm, as the accuracy of the snow values is +/- 1 cm.

It looks quite evident when comparing the weather data and cyclist counts that there is some correlation between them as seen in figure 1 displaying cyclist data on top of average weather, rain and snow amounts. The idea in the project is to use this correlation to predict the cyclist counts for future days based on available weather forecasts from FMI.

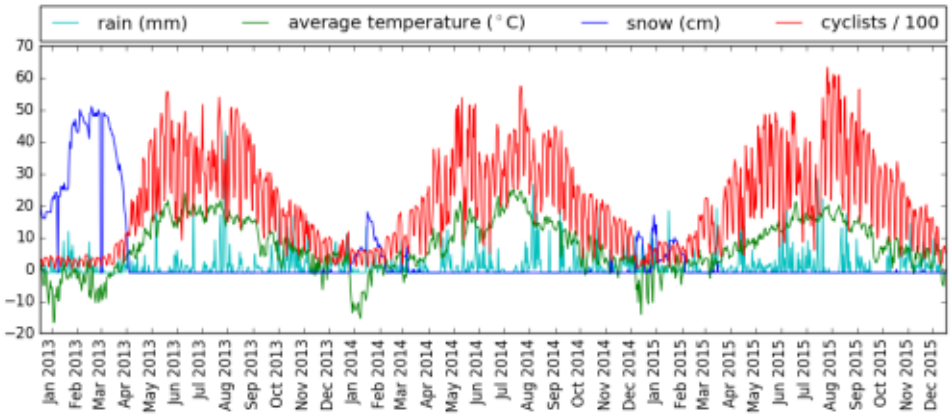

Figure 1: Cyclist counts on top of raw weather data.

Weather forecasts are provided only in hourly format and are missing depth of snow. For predictions, the most recent known depth of snow is used (yesterday) and an assumption is made that changes over night are not big enough to be relevant. The hourly temperatures and rain amounts are transformed into one daily value.

The predictions are collected for the next day every day at noon, but they are not saved as the data is not accurate. After making a prediction for tomorrow, yesterdays actual weather data is collected from FMI data service and saved to the training database to improve future predictions.

3 Linear classifier

A linear classifier algorithm is used to predict the cyclist counts based on weather data.

The accuracy of the model is tested using 90% of weather and cyclist pairs randomly selected within the data for training and the remaining 10% of pairs as a test set. The random selection is to make sure the test set contains values from all around the data period, instead of just the last few months. This gives us a better image of how accurate the model is within all times of year. All of the classifier versions have been tested with the same randomly selected data points, to make sure they are well comparable with each other.

The size of the test set is 115 data points and training set 1035 images in the figures and error values presented in this section. To evaluate the classifier error, a root-mean-square deviation (RMSD) has been calculated using these test and training sets. The formula used to calculate RMSD is

$$RMSD = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y)^2}{n}} \tag{1}$$

where n is the size of the test set, $\hat{y}$ the predicted cyclist count and y the actual cyclist count.

3.1 Version 1

The first version of the linear classification algorithm uses average temperature, rain and snow amounts as features and the cyclists counts as labels. It does not take into consideration anything more specific and considers each day and time of year similarly. The features are not scaled in any way.

The results for the first version of the project can be seen in figure 2. The test shows the predictions are not very accurate and overestimates cyclist counts with actual counts below 3000 cyclists and underestimates them with more than 3000 actual cyclists. RMSD for version 1 is 841.

For some test cases the predictions are negative values. Those have been leveled to 0, as no negative amount of cyclist can be using Baana at any moment. This practice has been used also with all the following versions.

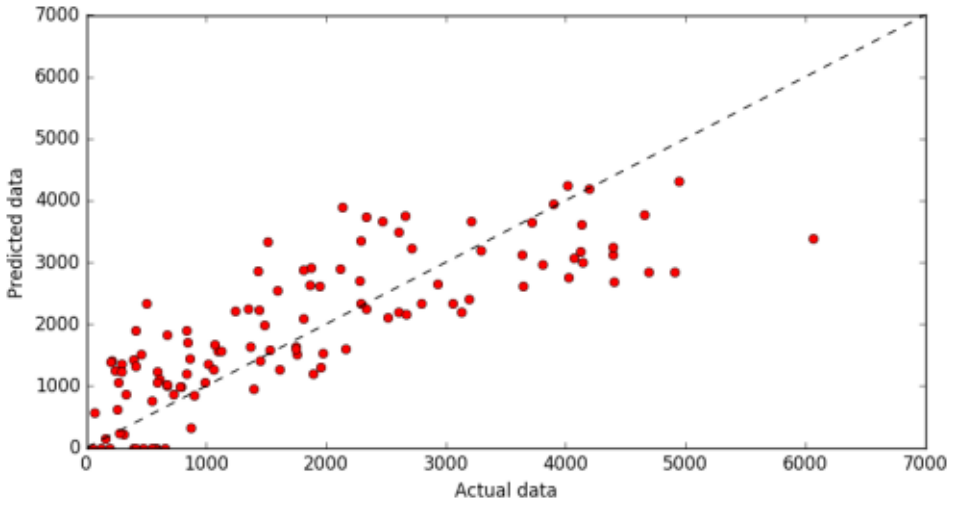

Figure 2: The test results for the first version of linear classifier used. Displays the actual cyclist counts for the test data with x axis and the predicted count within the y axis.

3.2 Version 2

With the second version of the linear classifier, seven separate classifiers have been created, one for each day of the week. When making predictions, a classifier for that day is used. This is because there is a noticeable drop in cyclist counts during weekends when there is less commuting, which might explain why the first version was not that good at being able to predict correctly very low or high cyclist counts.

The results can be seen in figure 3 and are centered closer to actual values. The RMSD for version 2 is 583, which makes it about 30% more accurate. Especially weekend predictions seem to be a bit more precise based on figure 3, while for the working week, The improvement is not as great. This might be due to working days having more variety in cyclist count depending for example on holidays and other factors that are not counted for. The training sets are also considerably smaller when dividing the data into 7 different classifiers.

Because not as much improvement was achieved as hoped, the classifier was changed in a way it only has two separate classifiers, one for working days and one for weekends. With the classifier version 2b seen in figure 4, the error was reduced some more and an RMSD value of 518 is achieved.

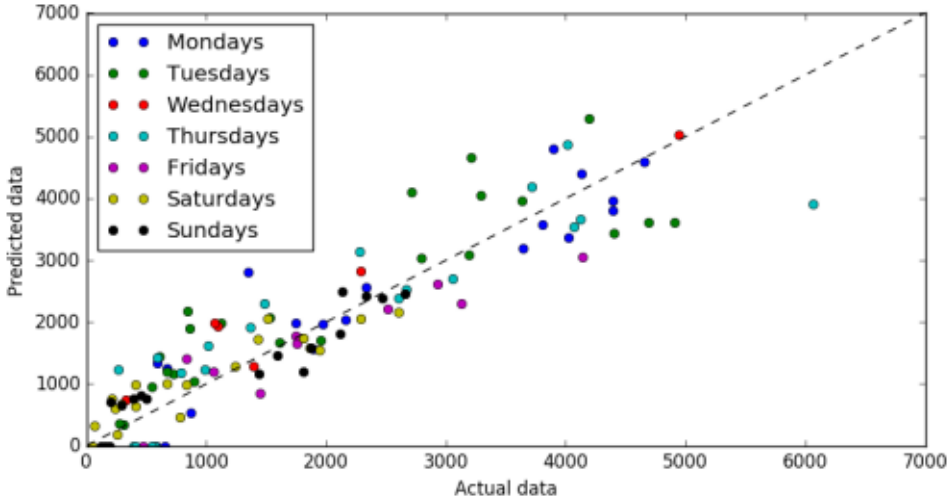

Figure 3: Second version of linear classifier used with each day of week with their separate colors.

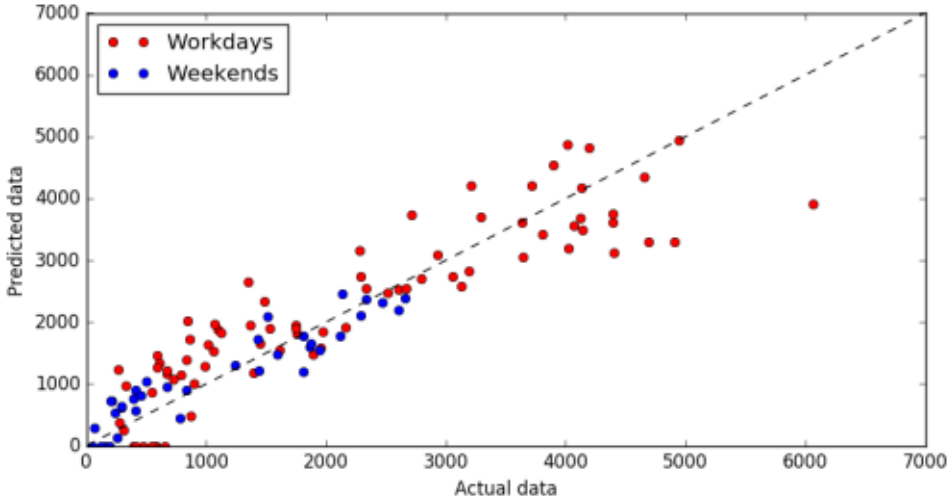

Figure 4: Version 2b with only workdays and weekends separated.

There still seems to be some of the same underlying problems with each of the classifiers as seen in figures 2, 3 and 4 where some spots (those where actual value is over 5000 for example) are predicted wrongly in the same way. The second classifier gets a bit closer to the target (diagonal line) with these data points but the visual representation of the data shows still the same error pattern.

Version 2b classifier has been predicting values from weather forecast on the project web service for two weeks. Observations received during that time are displayed in figure 5 and table 1. Table 1 displays the change in cyclist count since previous day with predicted and actual change, as well as the absolute difference (error) between the prediction and actual value. Cases where the sign of the change is different in the prediction opposed to the actual sign are highlighted in red as those are considered the biggest errors within the prediction. There are 4 cases like this within the two weeks test period, so overall the algorithm predicted correct gradient around 70% of time. The average difference between predicted and absolute change is 280.

Predicted change	Actual change	Absolute difference
197	-71	268
1019	722	297
-349	141	490
364	17	347
-181	-102	79
108	-128	236
-1347	-679	668
-86	-20	66
1228	593	635
97	82	15
73	-42	115
48	38	10
-196	-125	71
-1001	-383	618
312	27	285

Table 1: Predicted and actual change in cyclist count between predictions with version 2b classifier, as well as the absolute difference between the prediction and actual value.

Part of the problem with the classifier versions 1 and 2 is that they do not take advantage of the periodic nature of the data. After making several predictions using daily weather forecasts (figure 5) with the classifier 2 it is evident that the cyclist counts are quite close to each other on consecutive days and the algorithm should be able to take advantage of that. (The second weeks predictions are also most likely too high because the classifier has not taken into consideration it's skiing holiday week.)

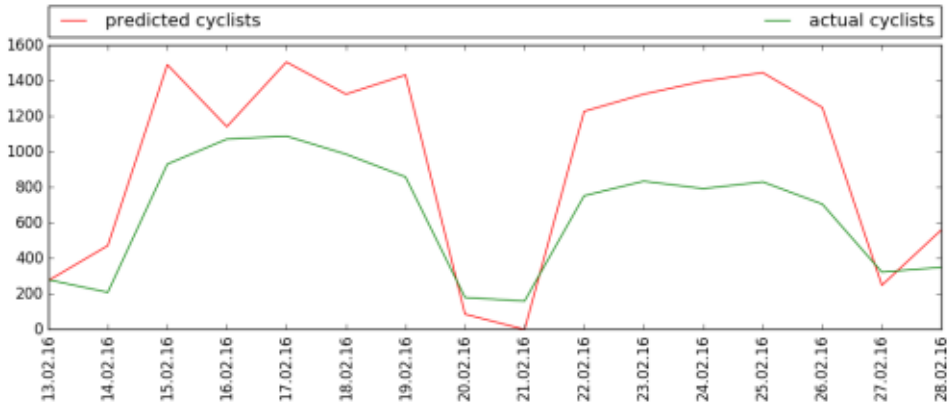

Figure 5: Predictions using classifier version 2 with real weather forecasts between 13th February and 28th February and the actual cyclist counts for the same days.

3.3 Version 3

The third version of the classifier takes into consideration some of the shortages noticed after making predictions with two weeks of weather forecasts using version 2. It also uses separate classifiers for weekends and working days, but in addition also knows the most recent cyclist count as well as the count from last week for the same day the prediction is made for.

The classifier does considerably better, with RMSD value of 377, over 55% improvement against the first version of the classifier and 27% improvement on version 2. The test data in figure 6 seems to have less extreme errors with high cyclist counts and less zero predictions.

After running the version 3 classifier against weather forecasts for 2 weeks, we can notice it overestimates the cyclist count less often, but predictions for especially Mondays and Tuesdays are too low. Predictions for Wednesdays to Fridays are quite accurate and weekends about as accurate as with the version 2 classifier as seen in figure 7. It almost seems as version 2 did better job with the predictions, even though it did overestimate the counts during workdays. Unfortunately no weather forecasts was saved until at the end of the project to be able to evaluate the classifiers side by side with the same forecasts.

The inaccuracy for Mondays and Tuesdays could be explained by the use of yesterdays cyclist count when making the prediction. This affects Tuesday as well, because the prediction for Tuesdays weather is made at noon

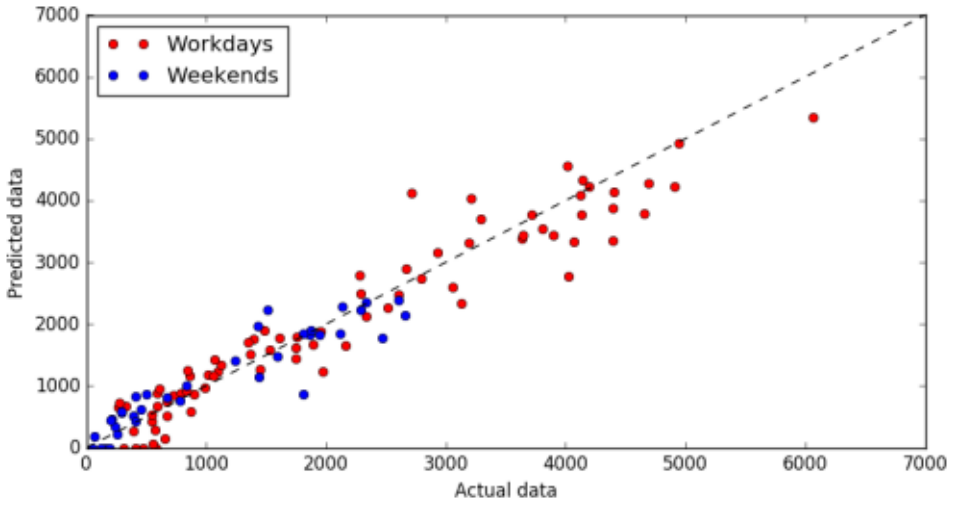

Figure 6: Plotted test results for version 3.

on Monday, where yesterdays cyclist count is Sundays count. Similarly for Monday the prediction is made at noon on Sunday when yesterdays cyclist count is Saturdays value. As we have seen previously, the counts for weekends are considerably lower than working days due to less commuting. The reason this does not affect the predictions for weekends in the same manner even though Fridays and Thursdays values are used is that we use a separate model for weekends and the model takes into account that yesterdays value is always higher than what it should predict. We can conclude that using yesterdays value as one of the features for the model was not a very good decision especially for the working days model. It would most likely be better to leave it out completely and only use last weeks value for the day we are predicting.

As for differences in predicted and actual changes in the cyclist counts, the results are quite similar with version 2 changes as seen in table 2. There are also 4 cases where the sign of the predicted change and actual change differ. Average difference between the changes is a bit lower at 232, adding for around 20% improvement.

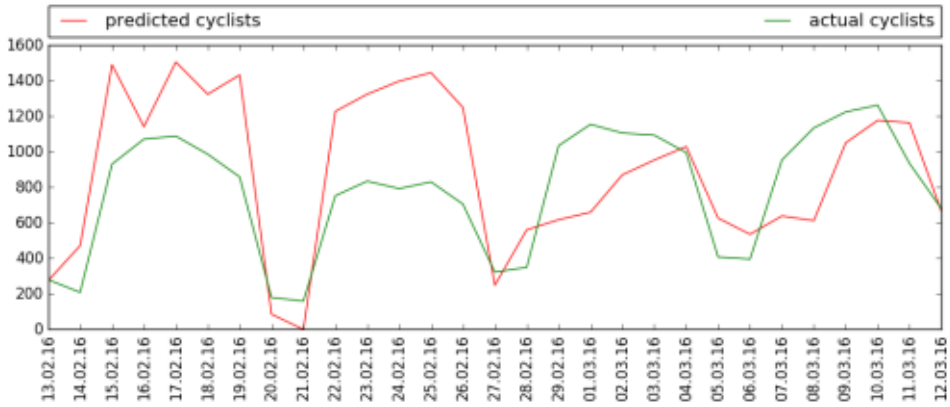

Figure 7: Predictions using classifier version 2 between 13th February and 28th February and classifier version 3 between 29th February and 12th March.

Predicted change	Actual change	Absolute difference
57	685	628
42	120	78
211	-49	260
83	-12	95
76	-98	174
-403	-588	185
-91	-12	79
103	558	455
-26	181	207
438	90	348
126	38	88
-13	-328	315
-499	-255	244
-159	-66	93

Table 2: Predicted and actual change in cyclist count between predictions with version 3 classifier, as well as the absolute difference between the prediction and actual value.

4 Displaying the results

Predictions for future cyclist counts are made available at http://baana-predictions.herokuapp.com/ website, which updates once a day with new predictions and displays results of past predictions and actual cyclist counts.

Updating predictions is done automatically with Heroku, GitHub and my own local server working together. The system is set up this way because

getting cyclist counts from Infracontrol [4] requires the count to be fetched before midnight daily and hosting cron jobs on Heroku is not a free service. This is why all the scheduling jobs is done on another server.

The way the system works is that the site in Heroku is simply a static page displaying prediction data read from a file. The files are stored in GitHub and updated daily by a cron job running twice a day on my server. One of the cron jobs reads the daily cyclist count just before midnight from Infracontrol and writes the count down locally. Another cron job runs daily just before noon, to get the most up to date forecast for tomorrows weather. The job runs the forecast through the trained model with most recent classifier version and writes down the prediction. It also checks yesterdays weather and writes it down to add to the training data, so the predictions will get more precise daily. After finishing all these tasks, it pushes the changes to GitHub and Heroku. This updates all new and past predictions on the website.

The website also displays past results as a plotted figure, updated each day at noon with new actual cyclist counts against past predictions, as well as weather data from the same period of time.

5 Conclusions and future work

In this project a linear model has been used for predicting how many cyclists are using Baana cycling route daily based on weather forecasts. Main focus has been in retrieving data from open sources, transforming it in a usable form for the machine learning algorithm and displaying and analyzing the results.

Three versions of the linear classifier was built, with the final version 3 algorithm using weather forecasts with average temperature, rain amount and snow height as well as yesterdays cyclist count and a cyclist count for last week for the day we are predicting as features. A different model was built for weekends and working days to account for less commuting on weekends. An observation was made that using yesterdays cyclist count as feature might disrupt predictions for Mondays and Tuesdays and leaving that feature out might improve prediction precision.

To evaluate the effect of weather within the cyclist counts, the predictions of version 3 classifier including weather data were compared with simply predicting last weeks value. The hypothesis is that if weather has no effect on

Actual	Predicted	Last week	Prediction error	Last week error	Day
1035	618	753	417	282	Mon
1155	660	835	495	320	Tue
1106	871	793	235	313	Wed
1094	954	831	140	263	Thu
996	1030	706	34	290	Fri
408	627	323	219	85	Sat
396	536	350	140	46	Sun
954	639	1035	315	81	Mon
1135	613	1155	522	20	Tue
1225	1051	1106	174	119	Wed
1263	1177	1094	86	169	Thu
935	1164	996	229	61	Fri
680	665	408	15	272	Sat
614	506	396	108	218	Sun
1625	901	954	724	671	Mon
1584	1036	1135	548	449	Tue

Table 3: Comparison between predicted cyclist counts with weather taken into consideration versus predicting last weeks count for the same day. Errors are calculated taking absolute value of subtracting the prediction or last weeks count from the actual cyclist count.

cyclist counts, then the counts should not significantly change from previous week and both classifiers should be similar in performance. The results are seen in table 3 and at first sight do not look promising for the model. During the 16 days the version 3 model was used for predictions, simply predicting last weeks value would have been the better choice in 10 of the cases. In total the error from version 3 classifier was 4401 while the error from selecting last weeks value was only 3659. Partly this is due to the high errors within Monday and Tuesday predictions caused by the poor selection of features. If Mondays and Tuesdays are ignored from the comparison, the total error for version 3 is 1380 and for last weeks value 1836 adding up to around 33% improvement when taking weather data into consideration. In this case the last weeks value predictor would have predicted closer to the correct count in only 4 of the 10 cases. This indicates that weather actually does have some effect on the counts.

To measure how significant the results are, a t-statistics test was made with a hypothesis, that cyclist counts are not correlated with previous days counts or weather but are instead random, following normal distribution

with arithmetic mean $\mu = 1000$ and sample standard deviation $\sigma^2 = 1000$, based on taking rated frequency distribution from the actual cyclist counts in table 3. The arithmetic mean of the predicted values is 815,5 and standard deviation 230,8. These can be used to calculate one-sample t-test

$$t = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \tag{2}$$

which for the predictions in table 3 gives $t \approx -3.198$. Level of significance is chosen to be $\alpha = 0.05$. The degree of freedom with the sample is $(n-1) = 15$ so we can see from t-table that the hypothesis can be considered valid if $-2.131 \leq t \leq +2.131$. This is not true as t is actually smaller than critical value -2.131, so we can reject the hypothesis that previous cyclist counts and weather does not affect cyclist counts. The sample size for both the evaluations done in this chapter is quite small, so the accuracy of the results should be taken as indicative at best.

More work needs to be done to get more precise predictions for each day of the week. Due to time constraints of this project a version 4 classifier was made but could not be evaluated. It contains the same features as version 3 only without yesterdays cyclist count. Another interesting experiment could be to use days lowest and highest temperatures instead of or in addition to average temperature. For a more precise classifier, it would also be important to consider school- and public holidays as seen with the drop of cyclist counts during skiing holiday.

References

[1] Ahmed, Farhana, Rose Geoff Jacob Christian: *Impact of weather on commuter cyclist behaviour and implications for climate change adaptation*. ATRF, 9, 2010. http://atrf.info/papers/2010/2010_Ahmed_Rose_Jacob.pdf.

[2] Finnish Meteorological Institute: *Open data*. https://en.ilmatieteenlaitos.fi/open-data. [Online; accessed 31-January-2016].

[3] Helsingin kaupunkisuunnitteluvirasto: *Number of cyclists travelling through baana*. https://www.avoindata.fi/data/en/dataset/baanan-pyorailijamaarat. [Online; accessed 31-January-2016].

[4] Infracontrol: *Baana real time measurement*. http://www1.infracontrol.com/cykla/barometer/barometer_fi.asp?system=helsinki&mode=day. [Online; accessed 18-February-2016].

[5] Infracontrol: *Traffic measurement*. https://www.infracontrol.com/en/its/traffic-measurement/. [Online; accessed 31-January-2016].

[6] Power, Nick and Snowdale Design: *Baanamittari*. https://baanamittari.fi/en/. [Online; accessed 31-January-2016].

[7] Sun, Hongyu, Liu Henry X. Xiao Heng He Racher R. Ran Bin: *Short-term traffic forecasting using the local linear regression model*, 2003. http://www.ltrc.lsu.edu/TRB_82/TRB2003-001580.pdf.