Pervasive Computing Exercise Report Assignment 1

Data Collecting

To outline the working process, this chapter describes how the data was collected.
I collected data on Altenbergerstraße in Linz and found that this street wasn't very suitable for the task. The traffic lights resulted in an irregular traffic flow, with vehicles driving in a narrow space, not well isolated from each other. In addition, the tram, together with traffic not directly involved in the measurement process, produced background noise.
So I looked at the map of Linz and found the best position. It turned out to be in the middle of Rechte Brückenstraße. It offers a lot of optimal conditions. First of all, it is the link between Urfahr and the industrial part of the city. So I assumed that there would be more lorries on the road than in other parts of the city. Furthermore, it is not the largest bridge leading to the industrial area. This means that the traffic tends to be more relaxed, but still active. There are three lanes in total, one for left to right and two for right to left. The latter includes a lane used only by buses. All three lanes have been considered.
As there are no traffic lights in the vicinity, the vehicles travelled at a constant speed and most of the time had a perfect distance between them.
In addition, background noise from construction sites and other city noise was greatly reduced on the bridge.
The wind was light. There was no rain or snow and the road was dry.

The date of the final recording was Tuesday 21.11.2023. The recording times were from 13:30 to 14:48 and 18:15 to 20:15. This resulted in a final recording span of over five hours.

The setting was as follows. The camera on the smartphone was facing the road. The microphone of the horizontal format of the smartphone pointed to the right. The position of the smartphone was approximately 1.2 m above the ground with a distance of approximately 1.5 m from the road, as required by the task instructions. The full setting is also shown in Figure 1.

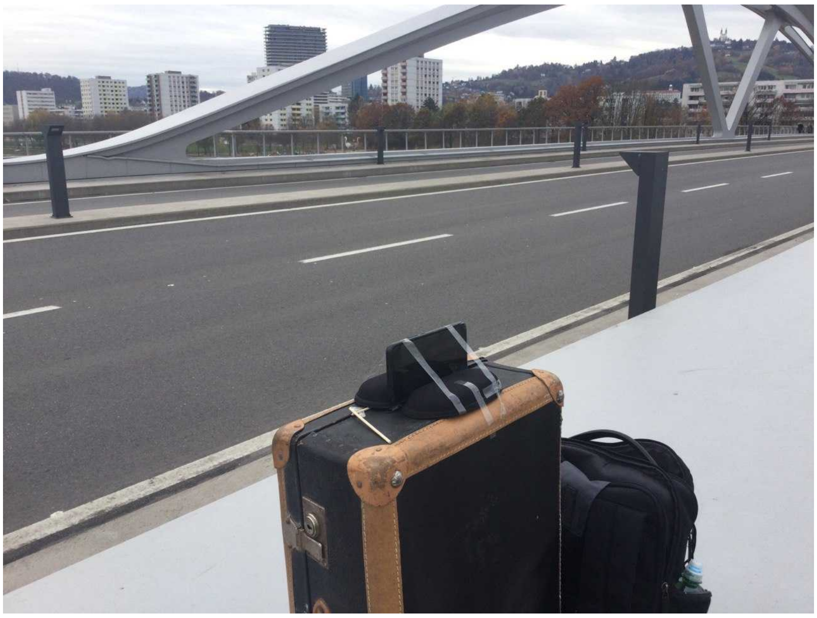

Figure 1: used setting showing the suitcase palced on an increase in metal. The smartphone points with the camera directly to the street and is fixed with tape.

Data Preparation

The 5 hours of video material was completely analysed by myself. To do this I used Plumber, a video editing software. I cut the videos into snippets/samples, where each sample contains one vehicle at a time. The snippet starts when the car enters the video and stops when the car leaves the filming window or when a second car enters the screen. The motivation for this approach was the image in the 'Acoustic Scene Analysis' tutorial slides, see also Figure 2.

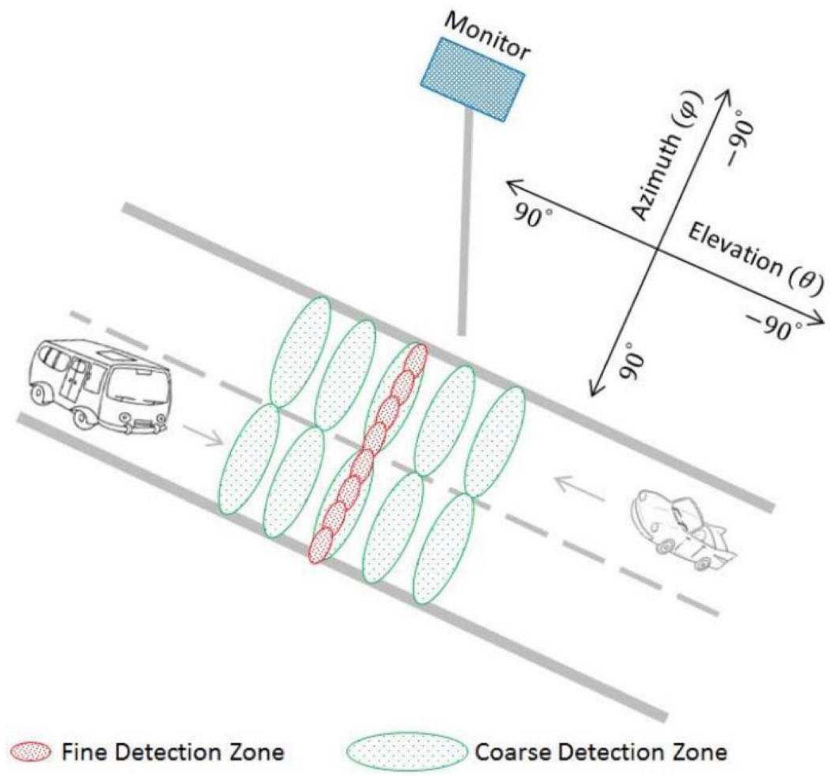

Figure 2: zones of detection presented in the exercise slides 'Acoustic Scene Analysis' on page 4.

In doing so, I defined the following class allocation:

- Light: mopeds, small cars
- Medium: normal size cars
- Heavy: trucks, SUVs, buses, vans

Finally, I sorted each video clip into light, medium and heavy folders. Within these folders there are folders called 'left' for left to right driving directions and 'right' for right to left driving directions. There are 2641 samples in total. All have been converted to wav files.

Table 1: data sample distribution				
	light	medium	heavy	Σ
left	336	941	430	1707
right	221	525	188	934
Σ	557	1466	618	2641

Table 1 shows that there are more left than right samples. In addition, the medium class is dominant. Downsampling of the major classes leads to a loss of information, so that the model may be negatively affected. Upsampling the small classes can help, but leads to overfitting. As will be discussed in the results chapters, different upsampling strategies were used and compared to the original data distribution.

Since ARFF files are ASCII compliant and can be used directly by WEKA, this format was created.

During the exercise it was pointed out that the noise in this task was so low that it could be treated as a constant. In addition, my location was quite well isolated from disturbing noise. So the noise was ignored.

Feature Selection

In the following two interesting feature types are introduced. Their settings are further explained in the results.

Mel-frequency Cepstral Coefficient (MFCC)
MFCC focuses on the spectral shape. Firstly, the Short-Time Fourier Transform (STFT) is mapped to a logarithmic scale. Therefore, MFCC accounts for the human perceptual features [1]. As the data measuring relays on the human perception about the mass of a car, MFCC becomes quite interesting.

Constant Q Transform (CQT)
The weak point of STFT is that it fails to represent the low and high frequencies as it is calculated in a linear frequency domain. CQT is determined with the help of a logarithmic frequency domain and honors low and high frequencies [2]. As CQT produces complex values their real-valued magnitudes are used for training. Here it would be interesting to see if CQT has some advantage compared to MFCC by considering low and high frequencies.

Hyper-parameter search process with intermediate results

As I believed, kNN (or in WEKA IBK) would only use one-vs-many class prediction I had a watch at MEKA, a program built up on WEKA but for multi-class classification. However, it turned out that MEKA didn‘t provide kNN. Therefore, I concentrated on the one-vs-many prediction. Light-vs-other vehicle was the focus on hyper-parameter search for models and features as it has a data imbalance in disadvantage for light vehicles such that it becomes a prediction challenge.
I built up three datasets:

- (a) : unmodified data samples
- (b) : upsample every second sample with light or heavy vehicle coming from the right as right is also in disatvantage

(c) : upsample every light and heavy sample coming from the right

Furthermore, most of the experiments were performed with MFCC. Only one hyper-parameter per test was modified. All trainings were performed with an ten-fold cross-validation. The mentioned results below are always **F-Measures** as it considers precision and recall at the same time.

The model notation is a follows.

NB : Naive Bayes
kNN : IBK, k-Nearest Neighbours
J48
MLP: Multi-Layer Perceptron

Light-vs-other vehicle MFCC

It turned out that (c) performed better than (b). Therefore, (b) was dropped in experiments.
At first step the hyper-parameters of MFCC were modified. I observed that increasing the sample rate from 22050 (default) to 44100 rose performance in kNN and J48 but lowered the score of NB. Next, the hop length was considered.

hop_length (best performances)	2048//4 (default)		330		800	
	(a)	(c)	(a)	(c)	(a)	(c)
NB	**.645**		.579		.592	
kNN		.789		.788		**.791**
J48		**.788**		.773		.775

In further experiments it was worked with hop_length=800 as it was in the first trial the best one (seems to be not the case according to table but this is only a re-run as WEKA died before successful saving). The input data sets struggled with the default setting of MLP, therefore MLP was finally involved in later experiments concentrating on the model hyper-parameters.

Next the question of n_mfcc appeared as it fixes the number of frequency bins of the individual DFTs.

n_mfcc	20 (default)		8		15	
	(a)	(c)	(a)	(c)	(a)	(c)
NB	**.712**		.673		.695	
kNN		**.829**		.813		.823
J48		**.791**		.788		.789

n_mfcc=20 scored the best what makes perfectly sense as more information about the frequencies is provided. However, we end up with more features which have to be processed. Therefore, I was interested in a lowering of n_mfcc while keeping the performance. Because for kNN and J48 the results are similar, you could use also n_mfcc=8. However, WEKA was surprisingly fast and therefore it was further worked with n_mfcc=20.

Final MFCC hyper-parameter setting: sample_rate=44100, hop_length=800, n_mfcc=20.

Next, the hyper-parameters of the models shift into focus.

NB (a): Changing the batch size (16, 32, 64, 100 (default), 256) and decimal places (2 (default), 4) had NO impact on the results. Setting the kernel estimator to True lowered the final score from **80.6** to 79.2. Therefore, the **default** setting seems to be the best. In the case of MFCC there was also the problem of the huge amount of unclassified samples (ca. 90.0038 %).

kNN: For (**c**) setting the number of neighbours from one to five lowered the results from 82.9 to 77.7. This is no surprise as (c) is the upsampled version of (a) and contains therefore duplicates which would end up exactly on their duplicate train sample in kNN space.
Therefore, also **(a)** was of interest as it had none. For (a) it was a positive impact to change the number of neighbours, by one neighbour a scores 57.9, with three it reaches 75.7, for five it falls back to 75.4. When setting the distance weighting from None to **1/distance** it scored with **three neighbours** even higher, **75.8** .

J48 c: Changing the confidence_factor (0.25 (default), 0.5), number of folds (3 (default), 10), batch size (16, 100 (default)), and binary split (False (default), True) had no impact. Therefore, the **default** setting seems to be the best, the score was always **79.1** .

MLP: (a): As I had troubles to run the MLP first I changed its hyper-parameters to have a good start: batch_size=4, lr=0.1, momentum=0, epoch number=10, hidden layer number=3. This resulted in a score of 78.88.
Next, I experimented with the number of epochs: **20 epochs→79.3**, 50 epochs→79.0, 100 epochs→78.7 . It turns out that 20 epochs seem to be already enough to score highly. This value is fixed and then the learning rate is lowered to 0.05 which resulted in a lower score 78.90. Thus, the learing rate stayed by 0.1 .
Also the number of hidden layers were of interest and resulted in the following: **six layers→79.33**, eight layers→78.6, 12 layers→78.7. Six layers beats the score of hidden layer number of three and is therefore used in the next act of changing the batch size: two batches→79.3, 32 batches→79.30. The batch size of four performed best.
(c): I tried out the same start setting as for (a). However, (c) scored slightly lower, **78.4** .
Finally, I ended up with the MLP hyper-parameter setting as given: 20 epochs, six hidden layers, momentum of zero, learing rate of 0.1, batch size of four.

Light-vs-other vehicle CQT

Using the hyper-parameter settings as they were found out above produced following table. Notice that the n_bins is equivalent to n_mfcc of MFCC. I modified it as follows.

n_bins	5		20	
	(a)	(c)	(a)	(c)
NB	.590	.590	**.614**	
kNN	.690	.767		**.771**
J48	.694	.755		**.776**
MLP	**.697**	.648	None	

Again, not surprisingly when we have more frequency bins the the CQT scores higher.

Furthermore, it can be observed that no model using CQT magnitude input can outperform any model using MFCC. However, it has the advantage that it was able to classify all samples with NB. For this reason, **CQT might be superior in NB compared to MFCC.**

Result overview

In this chapter I want to summarize all results. Notice that hyper-parameter search could be further extended. However, as I ran models over several days, it makes sense to finish the search. This can be also seen through the WEKA log files, which are only part-wise given as the computer died two times.
The classifier outputs are all stored in the appendix file.
CQT had only an advantage to MFCC in NB, therefore I consider here both input types separated. For the remaining models MFCC is taken.

In the evening before the deadline a fellow student told me that he was able to import multi-classes in WEKA. So far, I was only able to do binary prediction and assumed that it would be a WEKA issue as in the internet was stated that WEKA is not able to make multi-class predictions. However, I had then a rewatch of the content of my arff file and found the problem.
My error was: @attribute label_size integer\n → With that attribute design WEKA was not able to tell how many labels there would be. I changed it to: @attribute label_size {1,2,3}\n → This delivers the three possible classes of vehicle sizes:

1: light
2: medium
3: heavy

For directions I encoded:

1: left-to-right
2: right-to-left

Naive Bayes

Used hyper-parameter setting: default

light vs. medium vs. heavy

```
Time taken to build model: 2.4 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1022               38.6975 %
Incorrectly Classified Instances      1619               61.3025 %
Kappa statistic                          0.0445
Mean absolute error                      0.4084
Root mean squared error                  0.6367
Relative absolute error                103.348  %
Root relative squared error            143.2621 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,266    0,220    0,244      0,266    0,255      0,045    0,515     0,228     1
                 0,422    0,381    0,580      0,422    0,489      0,041    0,540     0,579     2
                 0,413    0,352    0,263      0,413    0,322      0,053    0,543     0,259     3
Weighted Avg.    0,387    0,340    0,435      0,387    0,400      0,045    0,536     0,430

=== Confusion Matrix ===

   a   b   c   <-- classified as
 148 224 185 |   a = 1
 319 619 528 |   b = 2
 139 224 255 |   c = 3
```

CQT (a)

```
Time taken to build model: 1.31 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1234               40.459  %
Incorrectly Classified Instances      1816               59.541  %
Kappa statistic                          0.0883
Mean absolute error                      0.3972
Root mean squared error                  0.6272
Relative absolute error                 93.9686 %
Root relative squared error            136.4209 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,306    0,208    0,335      0,306    0,320      0,101    0,558     0,304     1
                 0,435    0,352    0,533      0,435    0,479      0,084    0,553     0,519     2
                 0,445    0,350    0,314      0,445    0,368      0,087    0,568     0,306     3
Weighted Avg.    0,405    0,315    0,425      0,405    0,409      0,089    0,558     0,408

=== Confusion Matrix ===

   a   b   c   <-- classified as
 238 271 269 |   a = 1
 313 637 516 |   b = 2
 160 287 359 |   c = 3
```

CQT (c)

```
Time taken to build model: 2.28 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          77               2.9156 %
Incorrectly Classified Instances        66               2.4991 %
Kappa statistic                          0.0603
Mean absolute error                      0.3077
Root mean squared error                  0.5547
Relative absolute error               1274.8182 %
Root relative squared error            478.5234 %
UnClassified Instances                2498              94.5854 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,087    0,025    0,400      0,087    0,143      0,124    0,524     0,219     1
                 0,022    0,010    0,500      0,022    0,043      0,048    0,542     0,577     2
                 0,987    0,912    0,544      0,987    0,701      0,173    0,505     0,236     3
Weighted Avg.    0,538    0,485    0,507      0,538    0,404      0,126    0,520     0,340

=== Confusion Matrix ===

  a  b  c   <-- classified as
  2  0 21 |  a = 1
  3  1 41 |  b = 2
  0  1 74 |  c = 3
```

MFCC (a)

```
Time taken to build model: 3.46 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          94               3.082  %
Incorrectly Classified Instances        65               2.1311 %
Kappa statistic                          0.0844
Mean absolute error                      0.2743
Root mean squared error                  0.5228
Relative absolute error               1153.5308 %
Root relative squared error            462.7506 %
UnClassified Instances                2891              94.7869 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,103    0,023    0,500      0,103    0,171      0,163    0,526     0,265     1
                 0,026    0,008    0,500      0,026    0,049      0,067    0,548     0,506     2
                 0,989    0,897    0,596      0,989    0,744      0,208    0,507     0,267     3
Weighted Avg.    0,591    0,520    0,555      0,591    0,469      0,165    0,521     0,325

=== Confusion Matrix ===

  a  b  c   <-- classified as
  3  0 26 |  a = 1
  3  1 35 |  b = 2
  0  1 90 |  c = 3
```

MFCC (c)

For all models the F-Measure is relatively low. MFCC isn't even able to classify all samples. Looking at the confusion matrices of CQT, it becomes clear that Naive Bayes struggles with finding the difference between the classes. In slight advantage seems to be (c) due to its F-Measure. During training it become clear that Naive Bayes is highly scalable.

left-to-right vs. right-to-left

```
Time taken to build model: 3.49 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1501               56.8345 %
Incorrectly Classified Instances      1140               43.1655 %
Kappa statistic                          0.1011
Mean absolute error                      0.4318
Root mean squared error                  0.6541
Relative absolute error                 94.4404 %
Root relative squared error            136.8119 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,605    0,499    0,689      0,605    0,644      0,103    0,568     0,696     yes
                 0,501    0,395    0,410      0,501    0,451      0,103    0,581     0,402     no
Weighted Avg.    0,568    0,462    0,590      0,568    0,576      0,103    0,572     0,592

=== Confusion Matrix ===

    a    b   <-- classified as
 1033  674 |    a = yes
  466  468 |    b = no
```

CQT (a)

```
Time taken to build model: 1.26 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1764               57.8361 %
Incorrectly Classified Instances      1286               42.1639 %
Kappa statistic                          0.1495
Mean absolute error                      0.4223
Root mean squared error                  0.6461
Relative absolute error                 85.6752 %
Root relative squared error            130.1599 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,602    0,451    0,629      0,602    0,615      0,150    0,584     0,627     1
                 0,549    0,398    0,520      0,549    0,534      0,150    0,596     0,501     2
Weighted Avg.    0,578    0,428    0,581      0,578    0,579      0,150    0,589     0,571

=== Confusion Matrix ===

    a    b   <-- classified as
 1027  680 |    a = 1
  606  737 |    b = 2
```

CQT (c)

```
Time taken to build model: 2.71 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               3.3699 %
Incorrectly Classified Instances       143               5.4146 %
Kappa statistic                         -0.0035
Mean absolute error                      0.6164
Root mean squared error                  0.7851
Relative absolute error               1514.0277 %
Root relative squared error            546.6287 %
UnClassified Instances                2409              91.2154 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,041    0,046    0,600      0,041    0,077      -0,011   0,547     0,668     yes
                 0,954    0,959    0,374      0,954    0,537      -0,011   0,507     0,357     no
Weighted Avg.    0,384    0,388    0,515      0,384    0,250      -0,011   0,532     0,551

=== Confusion Matrix ===

   a   b   <-- classified as
   6 139 |   a = yes
   4  83 |   b = no
```

MFCC (a)

```
Time taken to build model: 1.2 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         129               4.2295 %
Incorrectly Classified Instances       136               4.459  %
Kappa statistic                          0.0246
Mean absolute error                      0.5132
Root mean squared error                  0.7164
Relative absolute error               1189.4123 %
Root relative squared error            485.9159 %
UnClassified Instances                2785              91.3115 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,050    0,024    0,700      0,050    0,093      0,068    0,554     0,588     1
                 0,976    0,950    0,478      0,976    0,642      0,068    0,505     0,443     2
Weighted Avg.    0,487    0,461    0,595      0,487    0,352      0,068    0,531     0,519

=== Confusion Matrix ===

   a   b   <-- classified as
   7 133 |   a = 1
   3 122 |   b = 2
```

MFCC (c)

Also in direction estimation the Naive Bayes for MFCC can't use most of the samples (ca. 91 %). CQT shows us that oversampling the underrepresented class b (right-to-left) only helps for MFCC. However, CQT has a higher F-Measure and seems to be better suited for the direction task.

kNN

Used hyper-parameter setting: number neighbours=3, distance weighting=1/distance

light vs. medium vs. heavy

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1512               57.2294 %
Incorrectly Classified Instances      1130               42.7706 %
Kappa statistic                          0.197
Mean absolute error                      0.3152
Root mean squared error                  0.4609
Relative absolute error                 79.785  %
Root relative squared error            103.699  %
Total Number of Instances             2642

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,276    0,083    0,471      0,276    0,348      0,240    0,673     0,382     1
                 0,804    0,649    0,607      0,804    0,692      0,175    0,612     0,639     2
                 0,288    0,096    0,478      0,288    0,360      0,234    0,673     0,424     3
Weighted Avg.    0,572    0,400    0,548      0,572    0,542      0,202    0,639     0,534

=== Confusion Matrix ===

   a    b    c   <-- classified as
  154  360   43 |    a = 1
  136 1180  151 |    b = 2
   37  403  178 |    c = 3
```

MFCC (a)

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1988               65.1803 %
Incorrectly Classified Instances      1062               34.8197 %
Kappa statistic                          0.4473
Mean absolute error                      0.245
Root mean squared error                  0.4076
Relative absolute error                 57.9485 %
Root relative squared error             88.6498 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,654    0,118    0,656      0,654    0,655      0,537    0,844     0,717     1
                 0,683    0,330    0,657      0,683    0,670      0,353    0,749     0,680     2
                 0,593    0,122    0,636      0,593    0,614      0,482    0,821     0,712     3
Weighted Avg.    0,652    0,221    0,651      0,652    0,651      0,434    0,792     0,698

=== Confusion Matrix ===

   a    b    c   <-- classified as
  509  233   36 |    a = 1
  228 1001  237 |    b = 2
   39  289  478 |    c = 3
```

MFCC (c)

Looking at the confusion matrix of MFCC (a) it turns out that knn works well on the medium class b. Upsampling leads here to the fact that duplicates end up in close positions to their related sample and increase the F-Measure.

left-to-right vs. right-to-left

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2513               95.1173 %
Incorrectly Classified Instances       129                4.8827 %
Kappa statistic                          0.8942
Mean absolute error                      0.0724
Root mean squared error                  0.204
Relative absolute error                 15.8342 %
Root relative squared error             42.6692 %
Total Number of Instances             2642

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,951    0,048    0,973      0,951    0,962      0,895    0,981     0,975     1
                 0,952    0,049    0,914      0,952    0,932      0,895    0,981     0,950     2
Weighted Avg.    0,951    0,049    0,952      0,951    0,951      0,895    0,981     0,966

=== Confusion Matrix ===

    a    b   <-- classified as
 1624   84 |    a = 1
   45  889 |    b = 2
```

MFCC (a)

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2923               95.8361 %
Incorrectly Classified Instances       127                4.1639 %
Kappa statistic                          0.9161
Mean absolute error                      0.0588
Root mean squared error                  0.1884
Relative absolute error                 11.9291 %
Root relative squared error             37.9495 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,936    0,013    0,989      0,936    0,962      0,918    0,989     0,985     1
                 0,987    0,064    0,924      0,987    0,954      0,918    0,989     0,978     2
Weighted Avg.    0,958    0,036    0,960      0,958    0,958      0,918    0,989     0,982

=== Confusion Matrix ===

    a    b   <-- classified as
 1598  109 |    a = 1
   18 1325 |    b = 2
```

MFCC (c)

It is worth to mention that (a) and (c) have F-Measures bigger than 0.9. It follows that the two classes of the direction task are well distinguishable even with a simpel model like the kNN.

Used hyper-parameter setting: number neighbours=5, distance weighting=1/distance:

light vs. medium vs. heavy

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1538               58.2355 %
Incorrectly Classified Instances      1103               41.7645 %
Kappa statistic                          0.1943
Mean absolute error                      0.3215
Root mean squared error                  0.4432
Relative absolute error                 81.3524 %
Root relative squared error             99.7168 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Clas
                 0,241    0,062    0,510      0,241    0,327      0,243    0,688     0,412     1
                 0,847    0,694    0,604      0,847    0,705      0,183    0,616     0,636     2
                 0,264    0,079    0,506      0,264    0,347      0,240    0,685     0,460     3
Weighted Avg.    0,582    0,416    0,561      0,582    0,541      0,209    0,647     0,548

=== Confusion Matrix ===

   a    b    c   <-- classified as
  134  386   37 |    a = 1
  103 1241  122 |    b = 2
   26  429  163 |    c = 3
```

MFCC (a)

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2075               68.0328 %
Incorrectly Classified Instances       975               31.9672 %
Kappa statistic                          0.4815
Mean absolute error                      0.2524
Root mean squared error                  0.3896
Relative absolute error                 59.6963 %
Root relative squared error             84.7388 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,630    0,090    0,705      0,630    0,665      0,561    0,855     0,748     1
                 0,769    0,367    0,660      0,769    0,711      0,405    0,759     0,678     2
                 0,567    0,084    0,707      0,567    0,629      0,521    0,822     0,729     3
Weighted Avg.    0,680    0,222    0,684      0,680    0,678      0,476    0,800     0,709

=== Confusion Matrix ===

   a    b    c   <-- classified as
  490  265   23 |    a = 1
  172 1128  166 |    b = 2
   33  316  457 |    c = 3
```

MFCC (c)

left-to-right vs. right-to-left

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2534               95.9485 %
Incorrectly Classified Instances       107                4.0515 %
Kappa statistic                          0.912
Mean absolute error                      0.0798
Root mean squared error                  0.1938
Relative absolute error                 17.4556 %
Root relative squared error             40.529  %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,961    0,043    0,976      0,961    0,968      0,912    0,985     0,981     1
                 0,957    0,039    0,930      0,957    0,944      0,912    0,985     0,961     2
Weighted Avg.    0,959    0,042    0,960      0,959    0,960      0,912    0,985     0,974

=== Confusion Matrix ===

    a    b   <-- classified as
 1640   67 |    a = 1
   40  894 |    b = 2
```

MFCC (a)

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2930               96.0656 %
Incorrectly Classified Instances       120                3.9344 %
Kappa statistic                          0.9207
Mean absolute error                      0.0658
Root mean squared error                  0.1821
Relative absolute error                 13.3541 %
Root relative squared error             36.689  %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,942    0,016    0,987      0,942    0,964      0,922    0,992     0,987     1
                 0,984    0,058    0,930      0,984    0,957      0,922    0,992     0,983     2
Weighted Avg.    0,961    0,034    0,962      0,961    0,961      0,922    0,992     0,985

=== Confusion Matrix ===

    a    b   <-- classified as
 1608   99 |    a = 1
   21 1322 |    b = 2
```

MFCC (c)

For both tasks, vehicle mass and direction, it is observable that even with a higher number of neighbours (five vs. three) the model performance keeps steady, although an increased number of neighbours leads to a simpler model.

J48

Used hyper-parameter setting: default

light vs. medium vs. heavy

```
Time taken to build model: 146.74 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1264               47.8607 %
Incorrectly Classified Instances      1377               52.1393 %
Kappa statistic                          0.131
Mean absolute error                      0.3488
Root mean squared error                  0.579
Relative absolute error                 88.2764 %
Root relative squared error            130.274  %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,311    0,202    0,292      0,311    0,301      0,107    0,561     0,238     1
                 0,581    0,473    0,605      0,581    0,593      0,108    0,554     0,589     2
                 0,387    0,198    0,373      0,387    0,380      0,186    0,604     0,305     3
Weighted Avg.    0,479    0,352    0,485      0,479    0,481      0,126    0,567     0,448

=== Confusion Matrix ===

   a   b   c   <-- classified as
 173 277 107 |   a = 1
 320 852 294 |   b = 2
 100 279 239 |   c = 3
```

MFCC (a)

```
Time taken to build model: 187.44 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1837               60.2295 %
Incorrectly Classified Instances      1213               39.7705 %
Kappa statistic                          0.3816
Mean absolute error                      0.2679
Root mean squared error                  0.5034
Relative absolute error                 63.3757 %
Root relative squared error            109.504  %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,647    0,165    0,573      0,647    0,607      0,464    0,755     0,486     1
                 0,595    0,297    0,650      0,595    0,622      0,301    0,658     0,610     2
                 0,572    0,164    0,556      0,572    0,564      0,404    0,731     0,464     3
Weighted Avg.    0,602    0,228    0,606      0,602    0,603      0,370    0,702     0,540

=== Confusion Matrix ===

   a   b   c   <-- classified as
 503 204  71 |   a = 1
 296 873 297 |   b = 2
  79 266 461 |   c = 3
```

MFCC (c)

Here the effect of upsampling becomes visible. The F-Measure is clearly higher (.481 vs. .603).

left-to-right vs. right-to-left

```
Time taken to build model: 41.48 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2110               79.894  %
Incorrectly Classified Instances       531               20.106  %
Kappa statistic                          0.5597
Mean absolute error                      0.2039
Root mean squared error                  0.4418
Relative absolute error                 44.595  %
Root relative squared error             92.3974 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,846    0,287    0,843      0,846    0,845      0,560    0,780     0,818     1
                 0,713    0,154    0,717      0,713    0,715      0,560    0,780     0,634     2
Weighted Avg.    0,799    0,240    0,799      0,799    0,799      0,560    0,780     0,753

=== Confusion Matrix ===

    a    b   <-- classified as
 1444  263 |    a = 1
  268  666 |    b = 2
```

MFCC (a)

```
Time taken to build model: 23.84 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2589               84.8852 %
Incorrectly Classified Instances       461               15.1148 %
Kappa statistic                          0.6963
Mean absolute error                      0.1565
Root mean squared error                  0.3836
Relative absolute error                 31.7458 %
Root relative squared error             77.2729 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,830    0,127    0,893      0,830    0,860      0,698    0,851     0,846     1
                 0,873    0,170    0,801      0,873    0,836      0,698    0,851     0,765     2
Weighted Avg.    0,849    0,146    0,852      0,849    0,849      0,698    0,851     0,810

=== Confusion Matrix ===

    a    b   <-- classified as
 1416  291 |    a = 1
  170 1173 |    b = 2
```

MFCC (c)

The tree is able to distinguish the direction classes quite well. Looking at the confusion matrices, the underrepresented class b has ca. the same amount of misclassifications in (a) and (c) although (c) has much more class b members. Therefore, the tree seems to have a slight bias towards the superior class in (a).

MLP

Used hyper-parameter setting: number of epochs=20, number hidden layers=6, momentum=0, learning rate=0.1, batch size=4

light vs. medium vs. heavy

```
Time taken to build model: 53.62 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1652               62.5521 %
Incorrectly Classified Instances       989               37.4479 %
Kappa statistic                          0.3185
Mean absolute error                      0.29
Root mean squared error                  0.4299
Relative absolute error                 73.3966 %
Root relative squared error             96.7218 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,402    0,083    0,566      0,402    0,470      0,365    0,766     0,492     1
                 0,799    0,542    0,648      0,799    0,716      0,275    0,662     0,663     2
                 0,414    0,089    0,587      0,414    0,486      0,371    0,746     0,539     3
Weighted Avg.    0,626    0,339    0,616      0,626    0,610      0,317    0,703     0,598

=== Confusion Matrix ===

    a    b    c   <-- classified as
  224  303   30 |    a = 1
  144 1172  150 |    b = 2
   28  334  256 |    c = 3
```

MFCC (a)

```
Time taken to build model: 91.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1939               63.5738 %
Incorrectly Classified Instances      1111               36.4262 %
Kappa statistic                          0.4179
Mean absolute error                      0.2794
Root mean squared error                  0.4192
Relative absolute error                 66.0958 %
Root relative squared error             91.1883 %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,599    0,139    0,596      0,599    0,597      0,459    0,810     0,622     1
                 0,689    0,362    0,638      0,689    0,663      0,327    0,723     0,668     2
                 0,574    0,099    0,676      0,574    0,621      0,502    0,819     0,680     3
Weighted Avg.    0,636    0,235    0,637      0,636    0,635      0,407    0,771     0,660

=== Confusion Matrix ===

    a    b    c   <-- classified as
  466  276   36 |    a = 1
  270 1010  186 |    b = 2
   46  297  463 |    c = 3
```

MFCC (c)

The F-Measures for MLP aren't too bad when considering its narrow network architecture and short training time. It was simply not feasible to train a deeper network on WEKA for the huge data amount.

left-to-right vs. right-to-left

```
Time taken to build model: 56.91 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2523               95.532  %
Incorrectly Classified Instances       118                4.468  %
Kappa statistic                          0.9021
Mean absolute error                      0.0653
Root mean squared error                  0.1904
Relative absolute error                 14.275  %
Root relative squared error             39.8141 %
Total Number of Instances             2641

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,968    0,067    0,963      0,968    0,966      0,902    0,983     0,988     1
                 0,933    0,032    0,941      0,933    0,937      0,902    0,983     0,976     2
Weighted Avg.    0,955    0,055    0,955      0,955    0,955      0,902    0,983     0,984

=== Confusion Matrix ===

    a    b   <-- classified as
 1652   55 |    a = 1
   63  871 |    b = 2
```

MFCC (a)

```
Time taken to build model: 38.24 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2924               95.8689 %
Incorrectly Classified Instances       126                4.1311 %
Kappa statistic                          0.9161
Mean absolute error                      0.0595
Root mean squared error                  0.1816
Relative absolute error                 12.0693 %
Root relative squared error             36.589  %
Total Number of Instances             3050

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,967    0,052    0,959      0,967    0,963      0,916    0,985     0,984     1
                 0,948    0,033    0,958      0,948    0,953      0,916    0,985     0,981     2
Weighted Avg.    0,959    0,044    0,959      0,959    0,959      0,916    0,985     0,983

=== Confusion Matrix ===

    a    b   <-- classified as
 1651   56 |    a = 1
   70 1273 |    b = 2
```

MFCC (c)

Also MLP performs well on the direction task. The diagonal entries of the confusion matrices (correct predictions) contain more than ten times more samples compared to the off-diagonal entries (failed predictions).

In summary, it was for the models much easier to make predictions for the direction task. Here the best performer is the kNN with five neighbours (c) with a F-Measure of .961 . In case of the vehicle mass task the models struggled to separate the classes. One reason could be that the classes aren't always easily separable. Further research could work with more than three classes in which different vehicle types are sorted to the classes by experts. The best perfoming model of this work is the kNN with five neighbours (.678).

References

[1] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. "Multimodal Music Information Processing and Retrieval: Survey and Future Challenges". In: 2019 InternationalWorkshop onMultilayerMusic Representation and Processing (MMRP). 2019, pp. 10–18. DOI: 10.1109/MMRP.2019.00012.

[2] Pei-Tse Yang et al. "Predicting Music Emotion by Using Convolutional Neural Network". In: HCI in Business, Government and Organizations. Ed. by Fiona Fui-Hoon Nah and Keng Siau. Cham: Springer International Publishing, 2020, pp. 266–275.