**Ugyen Wangchuck Institute**
**for Conservation and Environment**

**Slow Motion Projects**
**Dr Nicolas Perony**

### Technical report

# Machine learning approaches to automated behaviour classification from multimodal sensor data in Himalayan griffon vultures

## Summary

In this report, I illustrate an end-to-end machine learning workflow aiming to automatically characterise behavioural modes in Himalayan griffon vultures. I show that the use of ensemble methods for classification (specifically, gradient boosting models) can significantly improve the classification performance over that reported in the literature[1], with a near-perfect multi-class classification accuracy with balanced data. I discuss the utility of using multimodal data (acceleration and GPS) to identify behaviour, and demonstrate the importance of collecting a sufficient amount of training samples by generating synthetic data for sparse behavioural classes.

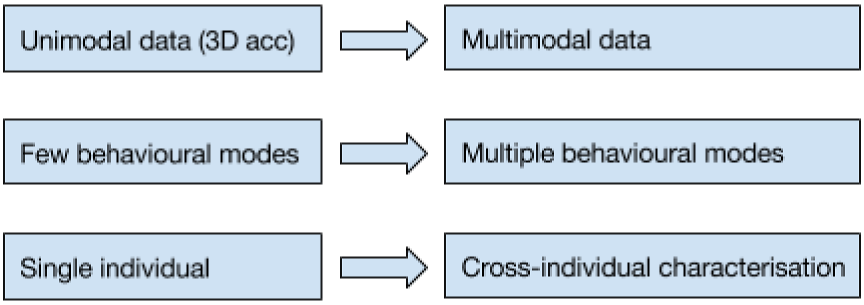
## Project description

The analysis was performed over a period of one week at UWICE in October 2016, based on labelled acceleration, GPS, and environmental data from two griffon vultures equipped with back-worn e-obs sensors.

## Project structure

In the following pages, I describe the sequential steps of the workflow I followed: (i) data cleaning, (ii) feature engineering, (iii) data exploration, (iv) feature selection, (v) model selection and validation.

I followed an approach of increasing complexity:



Unimodal data (3D acc) → Multimodal data

Few behavioural modes → Multiple behavioural modes

Single individual → Cross-individual characterisation

## Replicability

[1] Nathan, Ran, et al. "Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures." Journal of Experimental Biology 215.6 (2012): 986-996.

All the code necessary to replicate the analysis is at github.com/slowmotionprojects/uwice.

## Data cleaning

iPython notebook detailing workflow

Steps:

1. Tuples of CSV files containing acceleration and environmental variables were parsed and joined on `EventID`.
2. Duplicate records, identified by the same `EventID`, were deleted (first instance kept).
3. Non-matching records (based on `EventID`) in both files were deleted.
4. `Day` and `Time` columns were merged into a single column.
5. Records containing at least one invalid or absent value in any of the columns were deleted.
6. Records containing less than 40 acceleration measurements on each of the three axes for a burst were deleted.
7. The `Behavior` and `Behaviour` columns were harmonised into a single `state` column.

15853 and 4269 clean records were thus produced and exported to files `data_clean_Thang_Kaar_Baen_4185.csv` and `data_clean_Thang_Kaar_Dorje_4014.csv`, respectively.

## Feature engineering

iPython notebook detailing workflow

Steps:

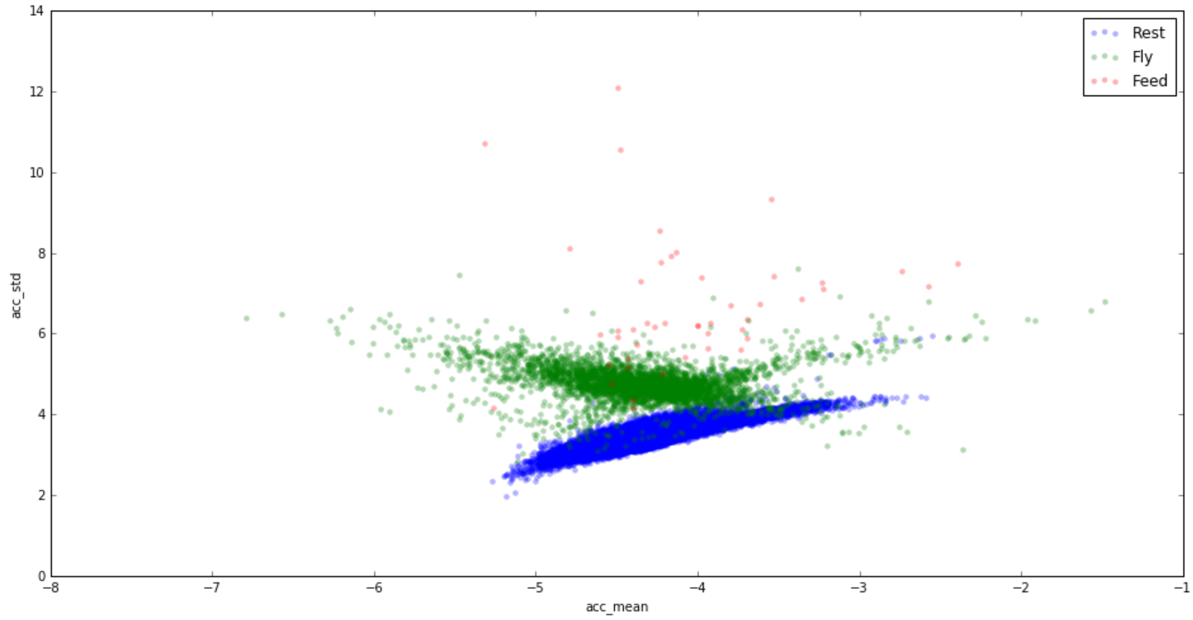
1. For each axis, acceleration values were standardised to zero mean and unit variance (calculated over all measurements for each animal), in order to compensate for the specificities of the sensor.
2. For each burst (40 acceleration measurements on each axis), composite acceleration variables were created: mean to characterise the general attitude during a burst, and standard deviation to characterise the attitude dynamics during a burst.
3. Altitude above ground was calculated as height above mean sea level minus SRTM elevation. To avoid outlier bias, altitudes under 0 m were set to 0 m, and altitude above 1000 m were set to 1000 m.
4. Temperature was converted to Celsius scale by subtracting 273.15 from the recorded values. To avoid outlier bias, temperature under -20°C were set to -20°C, and temperatures above +50°C were set to +50°C.
5. Other variables were imported without modification from the source data files, to obtain 17 variables for classification, with each record characterising a single burst: `EventID`, `dt`, `odba`, `groundspeed`, `pressuredelta`, `vvelocity`,

```
uvelocity,altitude,temperature,humidity,state,acc_x_mean,
acc_x_std,acc_y_mean,acc_y_std,acc_z_mean,acc_z_st.
```

## Data exploration

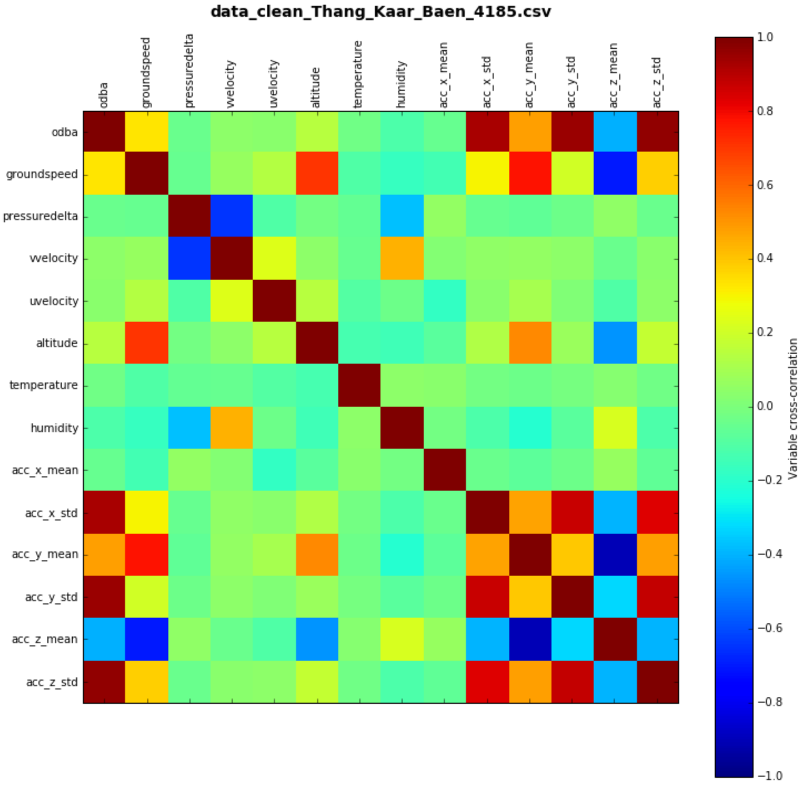
🔗 iPython notebooks (1, 2) detailing workflow (the first notebook shows a single individual only; the second notebook may not load in the GitHub web viewer due to heavy figures).

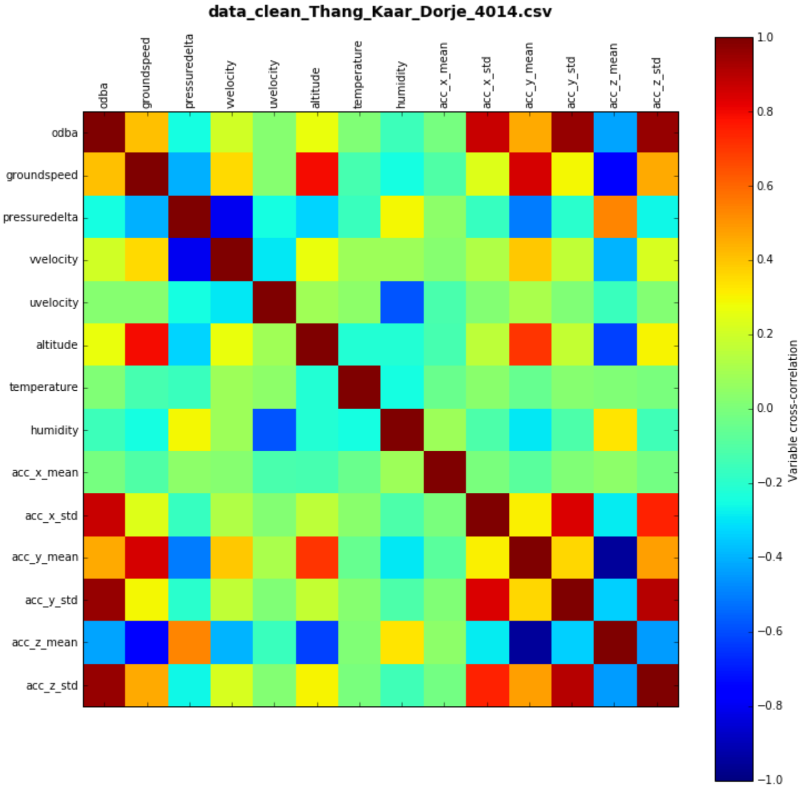
**Unimodal data (3D acceleration), 3 behavioural modes (Fly, Rest, Feed) only.**



Here, acc_mean is the mean of the mean acceleration over all three axes during a burst, and acc_std is the mean of the standard deviation of acceleration over all three axes during a burst. All three modes can be intuitively separated, however the Feed class shows a large dispersion into the Fly space, which will make it difficult to classify efficiently from acceleration only.

## Cross-correlation plots (multimodal data)

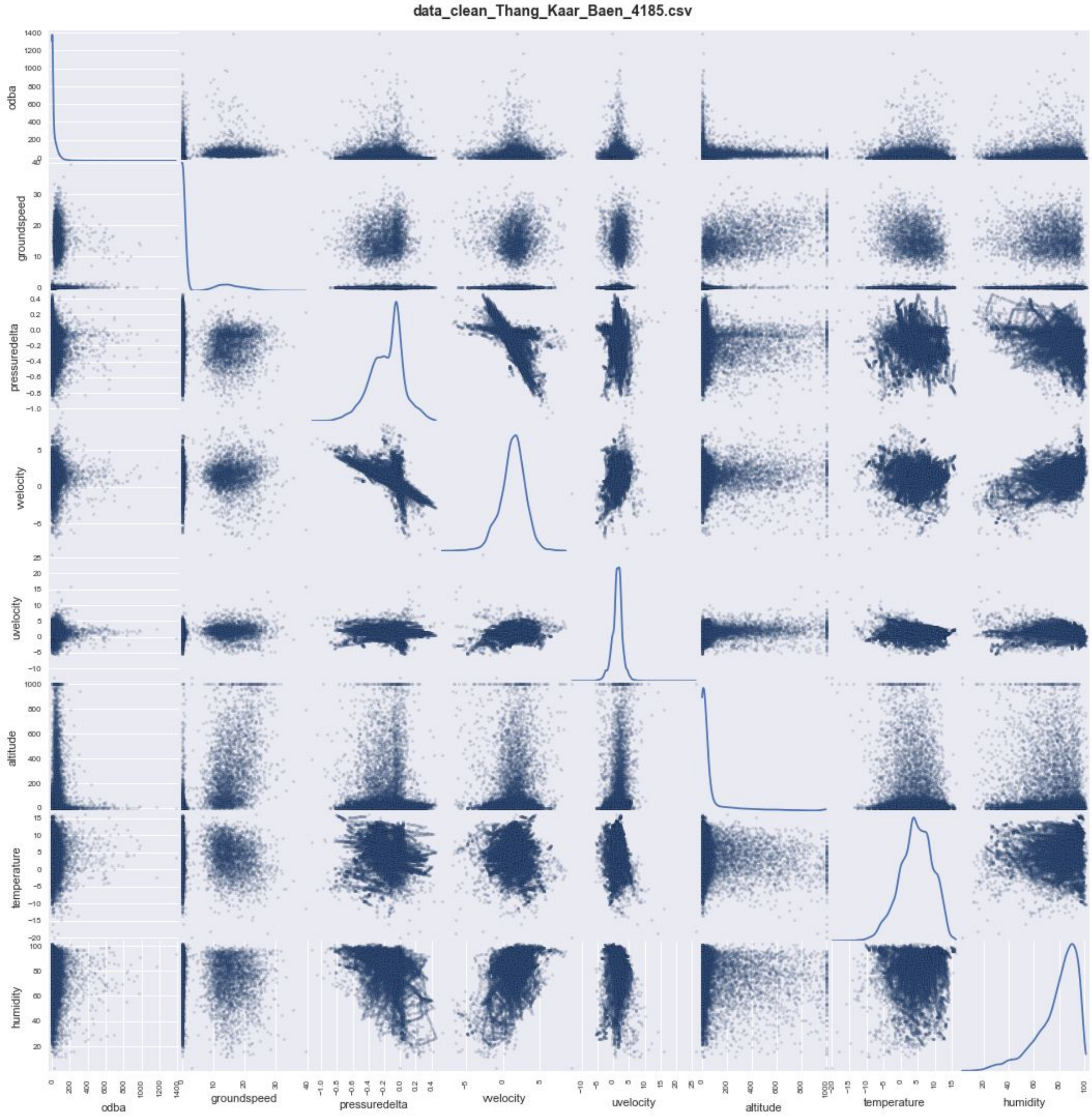


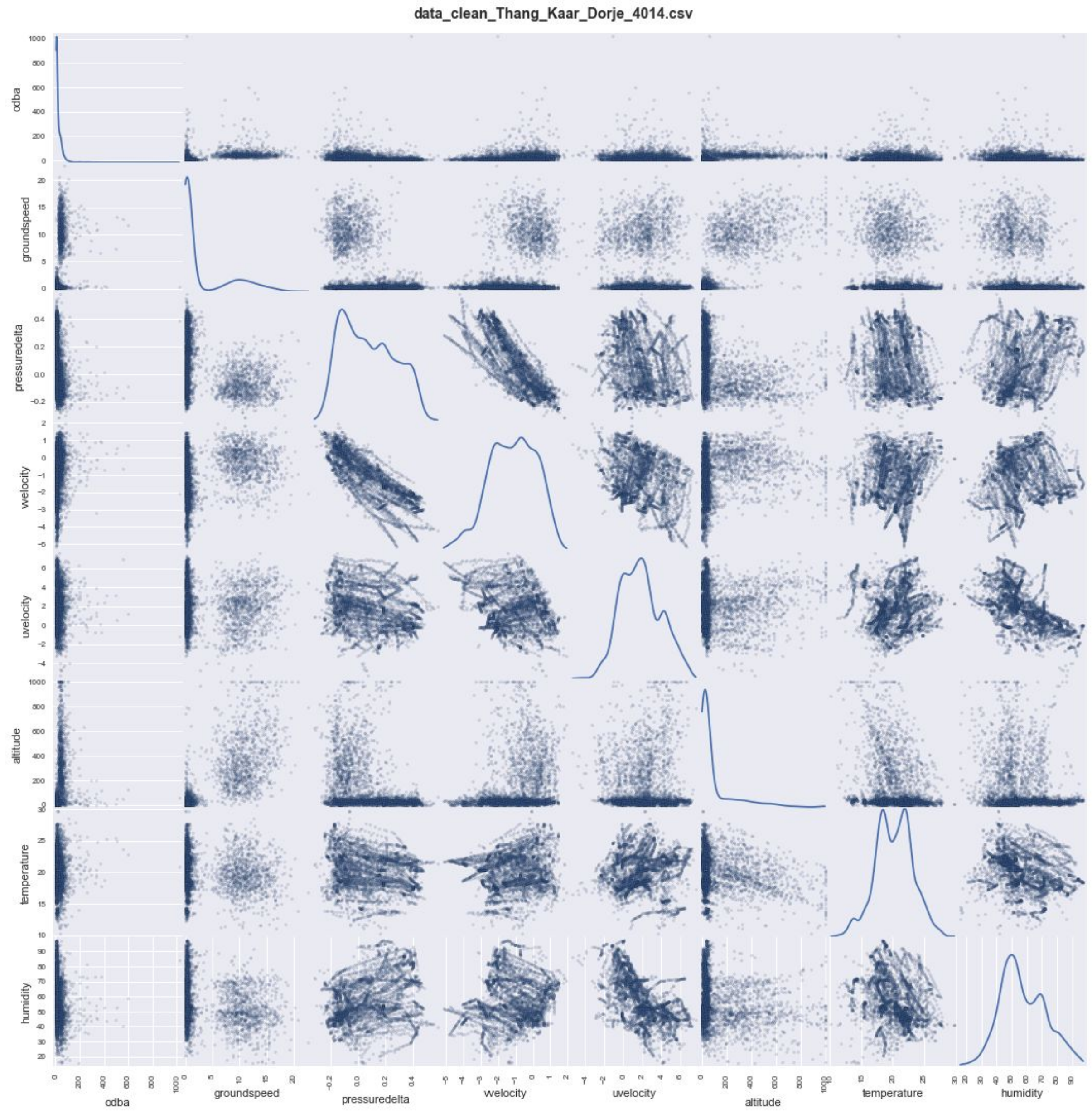


Dispersion (standard deviation) of acceleration on the X and Y axes are strongly cross-correlated, which makes sense due to the constraints of movement in the plane. Interestingly, acceleration on Z is strongly negatively correlated with acceleration on the other two axes, indicating that the corresponding movement/flight attitudes must be different. The Z axis also shows a negative correlation between mean and dispersion (standard deviation), showing that bouts of high acceleration (up or down) are relatively sustained. Finally, the mean acceleration on X seems to be completely decoupled from mean acceleration on other axes, which may indicate that this axis bears little information with regard to actual movement.

## Joint distribution plots (scatter matrices)

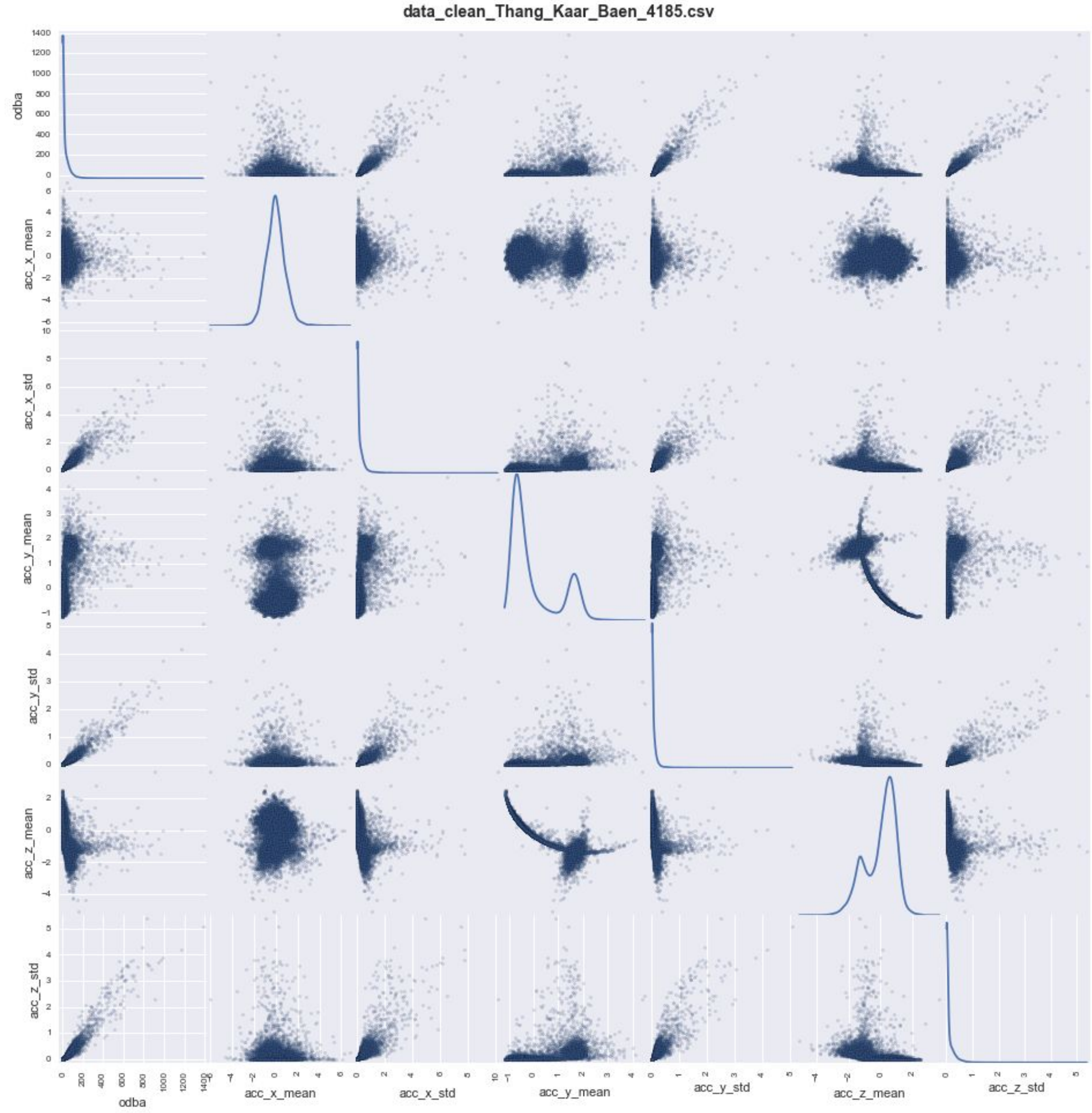
### 1. Environmental variables and ODBA

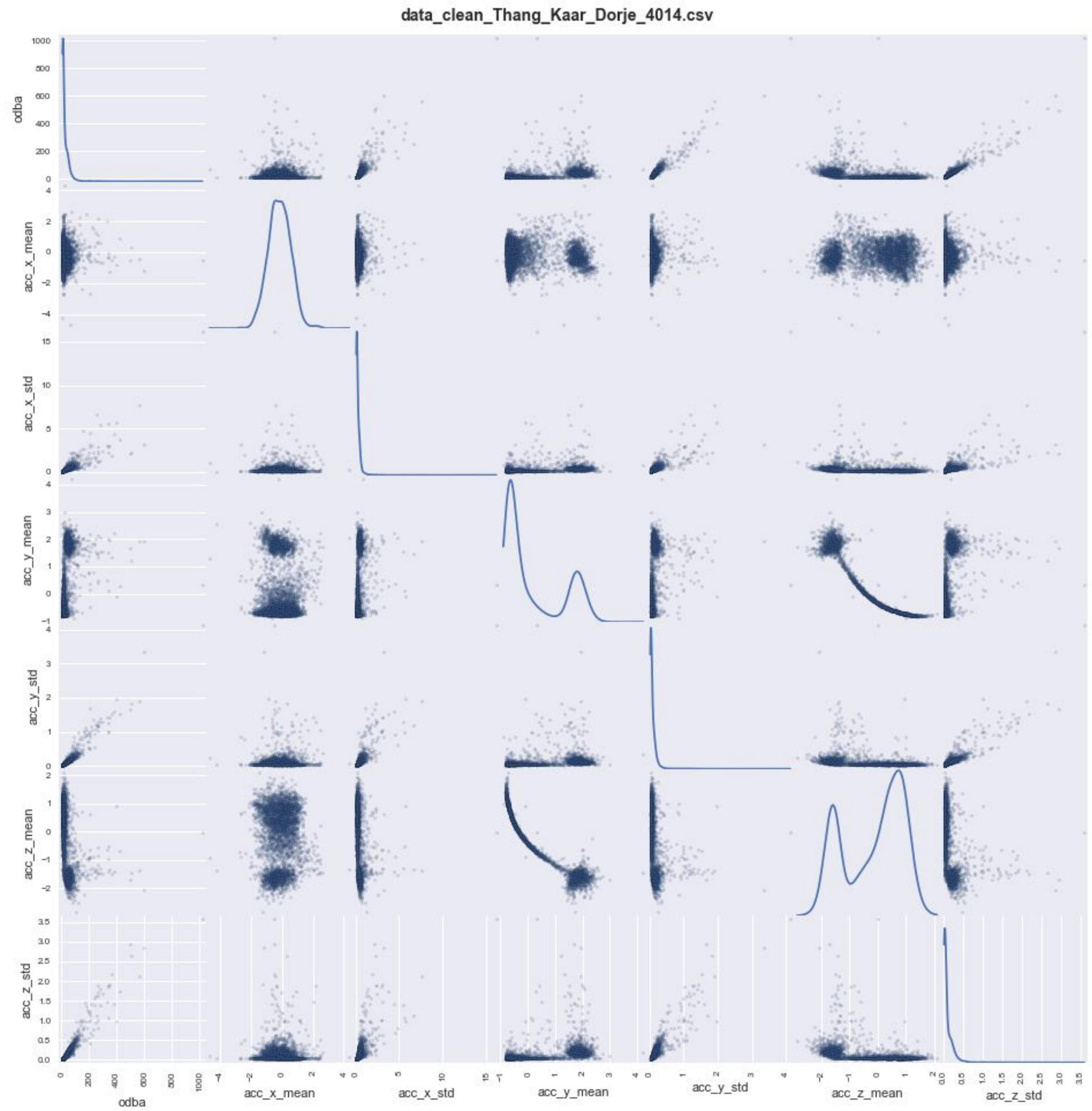




data_clean_Thang_Kaar_Dorje_4014.csv

## 2. Acceleration variables



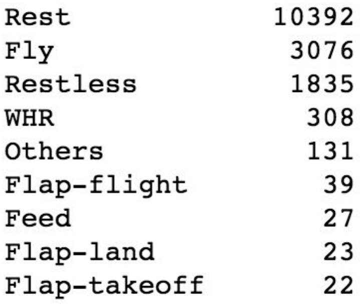


Interesting structure (clustering) can be found in the joint distributions, indicating favourable disposition towards classification. Some of the plots (e.g. UVelocity vs VVelocity) show streaks which may be artifacts of the measurement device. More puzzling are some of the nonlinear relationships between variables which should not, intuitively, show such relationships, e.g. the arc traced by mean Y acceleration vs mean Z acceleration. These may be signs of internal limitations of the sensors (e.g. limited frequency response). The interpretation of the other plots is left as an exercise to the reader :).

# Feature selection

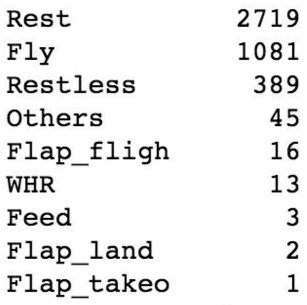
[iPython notebook detailing workflow]

Number of records labelled for each class, per individual:



```
Rest             10392
Fly               3076
Restless          1835
WHR                308
Others             131
Flap-flight         39
Feed                27
Flap-land           23
Flap-takeoff        22
```

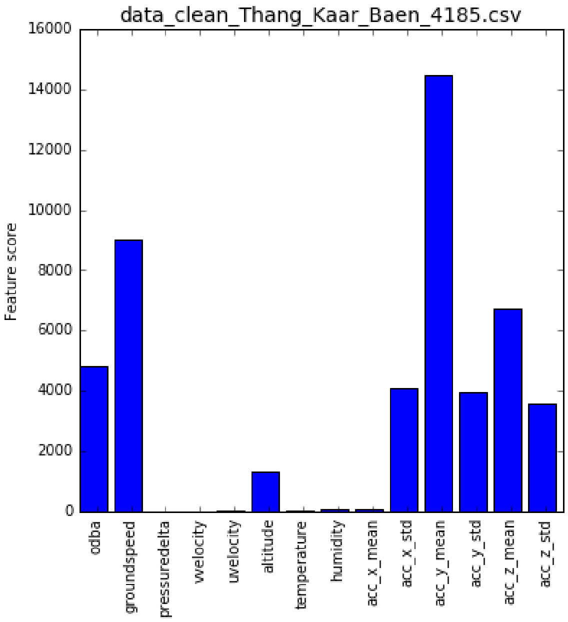
(Baen4185)

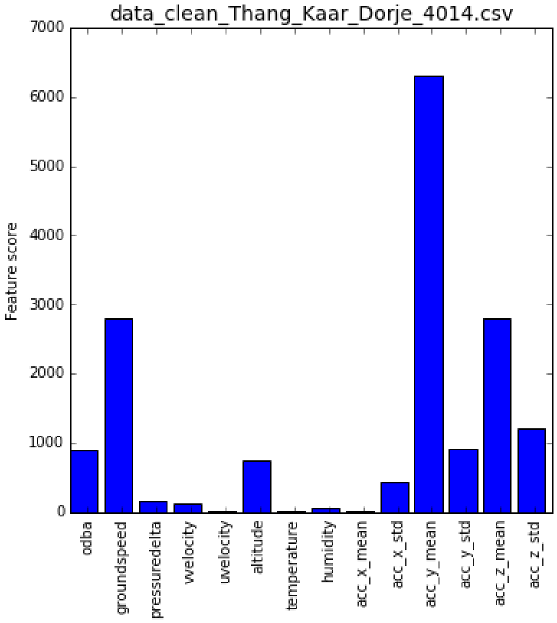


```
Rest          2719
Fly           1081
Restless       389
Others          45
Flap_fligh      16
WHR             13
Feed             3
Flap_land        2
Flap_takeo       1
```

(Dorje4014)

For most classes, the number of labelled records is so low that any model will have a very difficult time generalising to the concept of what constitutes an instance of, say, “Flap-land”. In order to mitigate the negative effect of class imbalance, and generally to improve the performance of the classification models, I performed univariate feature selection by computing the ANOVA F-value between each of the variables and the target (state) variable.





In both data sets, some variables (features) have a disproportionate weight in the determination of the target class (state). I kept the 8 variables with the highest score, namely all acceleration variables including ODBA but excluding mean acceleration on X, plus groundspeed and altitude. Note that since ODBA is a composite of all acceleration values, it could probably be removed from the feature set without penalising the model performance.

In addition, I also removed from the dataset all records belonging to unspecific classes such as “Others”, “Restless”, and “WHR”.

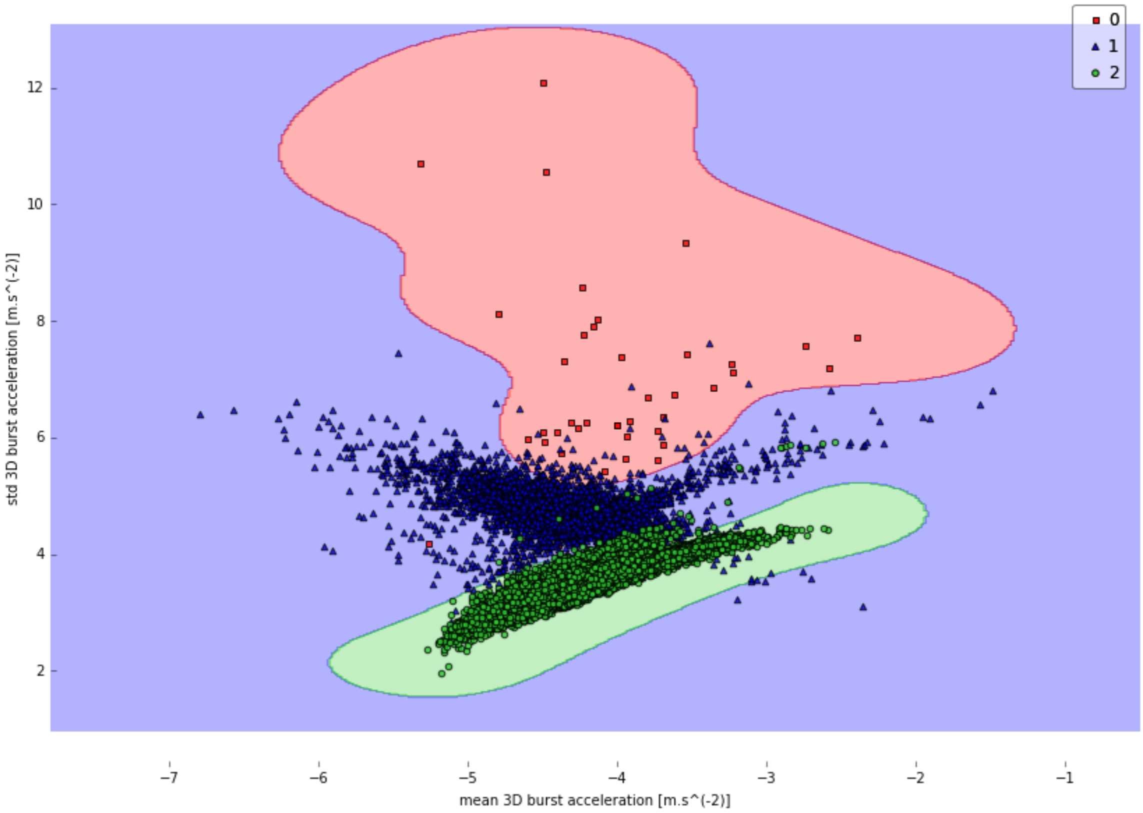
## Model selection and validation

iPython notebooks (1, 2) detailing worfklow (the first notebook presents a simple

model for a single individual and 3 classes only).

**Illustration: single individual, few classes**

I started by training a simple decision model (support vector machine with a radial kernel) on the data with only the three main classes (Rest, Fly, Feed), to quantify the ease of separation between these. The following is an illustration of the decision boundaries used by an instance of the simplified model (using only two variables) when trained to distinguish between these three classes (0=Feed, 1=Fly, 2=Feed). It appears that the classes are distinct (even in this simplified observation space), but there is also an overlap between the states, which indicates that a model would generalise better if given additional modes of measurement (e.g., GPS groundspeed to distinguish more easily between feed and fly).

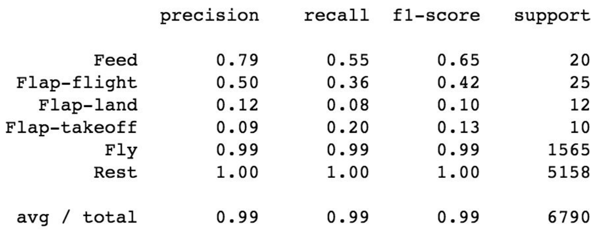


**Single animal, multiple classes**

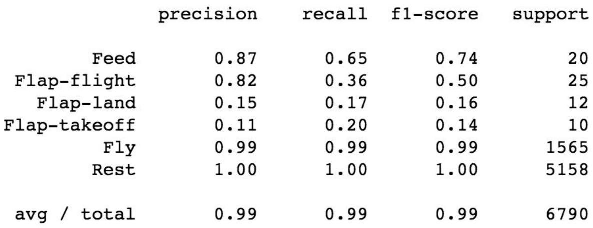
When adding more flight classes (Flap-flight, Flap-land, Flap-takeoff), the performance of a naive support vector machine collapsed. As an SVM works by building a hyperplane between the classes, the poor performance is probably linked to the absence of clear separation between the multiple classes, especially when dealing with so few examples. I introduced a much more powerful model class to deal with this difficulty: gradient boosting, which works by incrementally building decision trees focussing on the “hard” decisions, such as the separation between Fly and Feed in the above figure, and averaging the predictions of a large number of models.

**Performance gains from hyperparameter tuning**

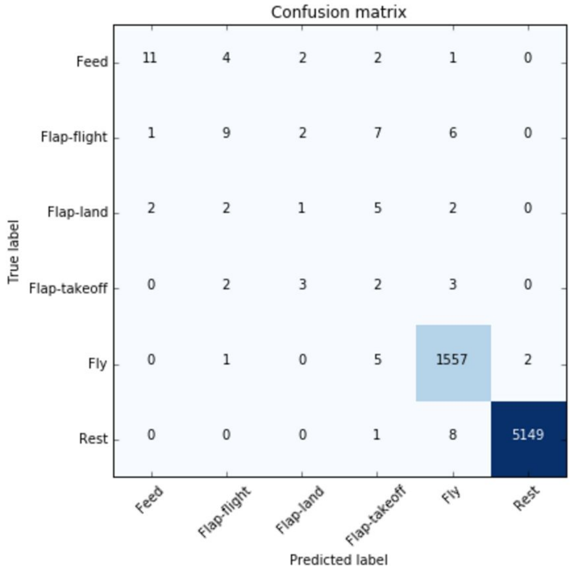
I trained a gradient booster on the larger data set (Baen4185), and then performed progressive hyperparameter tuning (exploration of the parameter space of the classifier) to improve its performance. Full details of the procedure are in the notebook linked above.



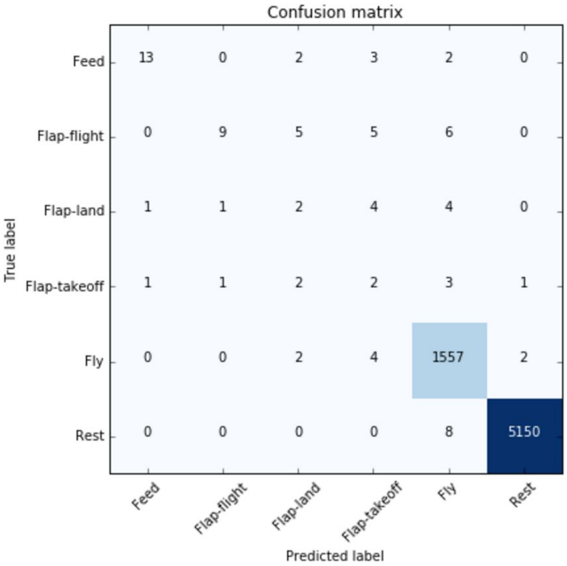
```
              precision    recall  f1-score   support

         Feed       0.79      0.55      0.65        20
  Flap-flight       0.50      0.36      0.42        25
    Flap-land       0.12      0.08      0.10        12
Flap-takeoff        0.09      0.20      0.13        10
          Fly       0.99      0.99      0.99      1565
         Rest       1.00      1.00      1.00      5158

  avg / total       0.99      0.99      0.99      6790
```



```
              precision    recall  f1-score   support

         Feed       0.87      0.65      0.74        20
  Flap-flight       0.82      0.36      0.50        25
    Flap-land       0.15      0.17      0.16        12
Flap-takeoff        0.11      0.20      0.14        10
          Fly       0.99      0.99      0.99      1565
         Rest       1.00      1.00      1.00      5158

  avg / total       0.99      0.99      0.99      6790
```



Untuned model



Tuned model

These results were obtained by k-fold cross-validation (k=10) over the full data set, and exhaustive grid search to select the best parameter combination. After this, the model was trained on one half of the data, and validated against the other half. As can be observed from both the F1-score and the confusion matrix, the impact of hyperparameter tuning on model performance is clear but not overwhelming.

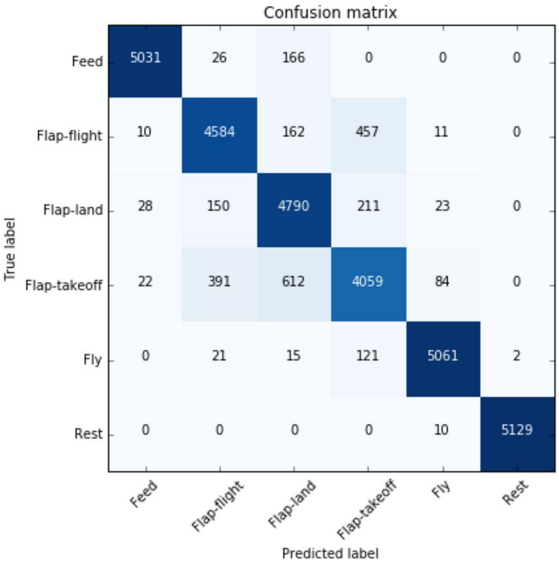
**The curse of imbalanced data: simulating performance gains on balanced classes**

Having imbalanced classes (many more observations resting than feeding, for example), can greatly penalise the performance of a model. Judging from the improvable performance of the model on the imbalanced data, I investigated the gain in classification accuracy that could be expected from more balanced data (for example if an animal could be labelled feeding as many times as it is labelled resting). Due to the low sample size, subsampling was not an option, so I oversampled by using two different methods: resampling with replacement (equivalent to observing many times the same rare events), or generating synthetic data by sampling from the estimated underlying distribution (using SMOTE[2]).

This procedure is equivalent to asking: how would the classification performance change if the rare observations could be made artificially more common, to balance the distribution of classes?

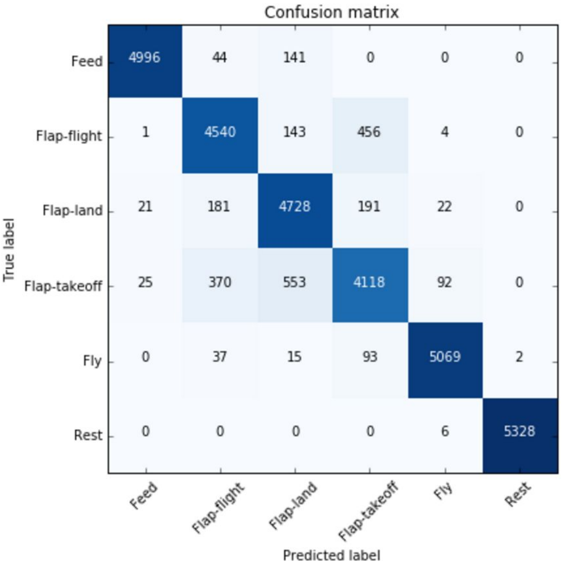
```
             precision    recall  f1-score   support

        Feed       0.99      0.96      0.98      5223
 Flap-flight       0.89      0.88      0.88      5224
   Flap-land       0.83      0.92      0.88      5202
Flap-takeoff       0.84      0.79      0.81      5168
         Fly       0.98      0.97      0.97      5220
        Rest       1.00      1.00      1.00      5139

 avg / total       0.92      0.92      0.92     31176
```

```
             precision    recall  f1-score   support

        Feed       0.99      0.96      0.98      5181
 Flap-flight       0.88      0.88      0.88      5144
   Flap-land       0.85      0.92      0.88      5143
Flap-takeoff       0.85      0.80      0.82      5158
         Fly       0.98      0.97      0.97      5216
        Rest       1.00      1.00      1.00      5334

 avg / total       0.92      0.92      0.92     31176
```



| | Feed | Flap-flight | Flap-land | Flap-takeoff | Fly | Rest |
|---|---|---|---|---|---|---|
| Feed | 5031 | 26 | 166 | 0 | 0 | 0 |
| Flap-flight | 10 | 4584 | 162 | 457 | 11 | 0 |
| Flap-land | 28 | 150 | 4790 | 211 | 23 | 0 |
| Flap-takeoff | 22 | 391 | 612 | 4059 | 84 | 0 |
| Fly | 0 | 21 | 15 | 121 | 5061 | 2 |
| Rest | 0 | 0 | 0 | 0 | 10 | 5129 |

Resampling with replacement



| | Feed | Flap-flight | Flap-land | Flap-takeoff | Fly | Rest |
|---|---|---|---|---|---|---|
| Feed | 4996 | 44 | 141 | 0 | 0 | 0 |
| Flap-flight | 1 | 4540 | 143 | 456 | 4 | 0 |
| Flap-land | 21 | 181 | 4728 | 191 | 22 | 0 |
| Flap-takeoff | 25 | 370 | 553 | 4118 | 92 | 0 |
| Fly | 0 | 37 | 15 | 93 | 5069 | 2 |
| Rest | 0 | 0 | 0 | 0 | 6 | 5328 |

SMOTE

The resampling with replacement method shows higher performance but it is more artificial, since it constrains the model to increase the weight of single observations. The SMOTE method is more natural, and does not lead to a large deterioration in performance.

With both resampling methods, we observe that the performance of the optimised model is very high, with the classification errors more evenly distributed across the dataset (as we've artificially penalised the model for class imbalance), and the confusion matrix helps to see which classes are still hard to tell apart. The 3 flap flight classes (especially take-off and land) are the only ones that regularly get mixed up; on other classes, the classification is close to 100%, demonstrating the advantage of having a more balanced data set.

The results above indicate that, **for a single individual, and provided there is a sufficient number of labelled instances available for each class, it is possible to train**

[2] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

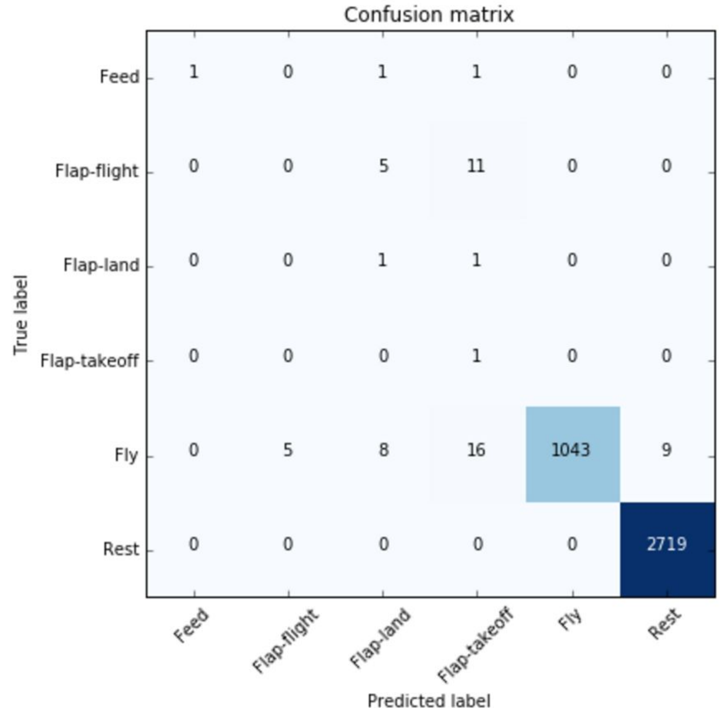
**a machine learning classifier to reach near-perfect accuracy** (note that these models were not further tuned after balancing the data set, which would have led to even higher accuracy).

**Cross-individual model validation: transferring learned behavioural concepts from one individual to another**

Beyond single-animal classification, I sought to know if the model could extrapolate to what it means for a griffon vulture to perform different behaviours; in other words, is it possible to teach the model to generalise?

I took the model trained on the first individual (Baen4185), and applied it without retraining to the second individual (Dorje4014). Note that this is akin to teaching a computer to recognise hair on a person's face by showing it dark hair only, and then testing its accuracy in identifying blonde hair, hence a very difficult task.

```
              precision    recall  f1-score   support

         Feed       1.00      0.33      0.50         3
  Flap-flight       0.00      0.00      0.00        16
    Flap-land       0.07      0.50      0.12         2
 Flap-takeoff       0.03      1.00      0.06         1
          Fly       1.00      0.96      0.98      1081
         Rest       1.00      1.00      1.00      2719

  avg / total       0.99      0.99      0.99      3822
```



| True label \ Predicted label | Feed | Flap-flight | Flap-land | Flap-takeoff | Fly | Rest |
|---|---|---|---|---|---|---|
| Feed | 1 | 0 | 1 | 1 | 0 | 0 |
| Flap-flight | 0 | 0 | 5 | 11 | 0 | 0 |
| Flap-land | 0 | 0 | 1 | 1 | 0 | 0 |
| Flap-takeoff | 0 | 0 | 0 | 1 | 0 | 0 |
| Fly | 0 | 5 | 8 | 16 | 1043 | 9 |
| Rest | 0 | 0 | 0 | 0 | 0 | 2719 |

Cross-individual performance of the gradient boosting model without retraining

Remarkably, much of the classification accuracy on the classes outside of flap flight is preserved. Generally, the similar performance on these classes (mainly fly and rest) for the two individuals shows that there is high potential for generalisation of behavioural mode learning. When attempting to solve this problem, much attention should be paid to the standardisation of features, so that the specificities of each recording setup (tag calibration etc.) do not affect the performance of the classifier.