

---

# Ensemble Learning Applied to Classify GPS Trajectories of Birds into Male or Female

---

**Dewan Fayzur**  
DevScope S.A.  
Porto, Portugal  
fayzur20@gmail.com

## Abstract

We describe our first-place solution to the Animal Behavior Challenge (ABC 2018) on predicting gender of bird from its GPS trajectory. The task consisted in predicting the gender of shearwater based on how they navigate themselves across a big ocean. The trajectories are collected from GPS loggers attached on shearwaters' body, and represented as a variable-length sequence of GPS points (latitude and longitude), and associated meta-information, such as the sun azimuth, the sun elevation, the daytime, the elapsed time on each GPS location after starting the trip, the local time (date is trimmed), and the indicator of the day starting the from the trip. We used ensemble of several variants of Gradient Boosting Classifier along with Gaussian Process Classifier and Support Vector Classifier after extensive feature engineering and we ranked first out of 74 registered teams. The variants of Gradient Boosting Classifier we tried are CatBoost (Developed by Yandex), LightGBM (Developed by Microsoft), XGBoost (Developed by Distributed Machine Learning Community). Our approach could easily be adapted to other applications in which the goal is to predict a classification output from a variable-length sequence.

# 1 Introduction

The Animal Behavior Challenge was organized by the 2018 Symposium on Systems Science of Bio-Navigation <sup>1</sup>, sponsored by Technosmart <sup>2</sup> and proposed as a CodaLab competition <sup>3</sup>. It consisted in classifying the gender of shearwater based on trajectories (latitude and longitude) and some meta-information associated to each shearwaters' trip. Such prediction models could help to understand shearwater more efficiently and how they navigate themselves, like male and female shearwater could use different trajectories along the way of trip.

The training dataset is composed of all the GPS trajectories of 631 streaked shearwaters (326 male and 305 female) breeding on Awashima Island, Japan. Each datapoints in the training dataset representing a complete bird trip and being composed of the following attributes <sup>4</sup>:

- longitude
- latitude
- sun azimuth: clockwise from the North
- sun elevation: upward from the horizon
- daytime: 1 being day, or 0 being night
- elapsed time: after starting the trip
- local time: only time with a format (hh:mm:ss)
- days: days after the trip starts

In the competition setup, the testing dataset is composed of all the GPS trajectories of 275 streaked shearwaters. In the Development Phase of the competition 10% of the submission labels are randomly modified to report score. In the Final Phase of the competition, all the submission are recalculated for the final ranking.

Our approach uses extensive feature engineering prior to ensemble learning. Section 2 describes feature engineering techniques. Section 3 introduces our winning model, which is based on a ensemble learning architecture. Section 4 compares and analyses our various models quantitatively and qualitatively on the competition dataset set. The source code of our first-place solution can be found online <https://github.com/dfayzur/Animal-Behavior-Challenge-ABC2018>.

## 2 Feature Engineering

### 2.1 Basic New Features

We first created velocity, acceleration, distance features for each of the GPS points from the gives dataset. At this time we have 7 key features for each GPS points to work with, such features are: velocity, acceleration, distance, longitude, latitude, azimuth, and elevation. We also created the differences of features at time  $t$  to the next point at time  $t+1$ . We call these features as *delta* of velocity, longitude, latitude, azimuth, and elevation. From these 12 features, we took quintiles at 0%, 5%, 10%, 20%, 25%, 30%, 40%, 50%, 60%, 70%, 75%, 80%, 90%, 95%, and 100% of them. In addition, we also calculated average, minimum, and maximum of these 12 features. All these operations are applied on each sequence of GPS trajectories for each bird individually. We also calculated the number of times a velocity exceeds to values (average and quintiles at 5%, 10%, 15%, 25%, 50%, 75%, 80%, 85%, 90%, 95%, and 99%) calculated over all GPS trajectories combined. Finally we took first 5 longitude and latitude values of each birds' trajectories. We also included Principal component analysis (PCA) on longitude, latitude, azimus, elevation, and velocity of individual birds' and added them to the final features list.

---

<sup>1</sup><http://navi-science.org/2017/09/19/symposium-on-systems-science-of-bio-navigation-2018/>

<sup>2</sup><http://www.technosmart.eu/index.php>

<sup>3</sup><https://competitions.codalab.org/competitions/16283>

<sup>4</sup>[https://competitions.codalab.org/competitions/16283#participate-get\\_data](https://competitions.codalab.org/competitions/16283#participate-get_data)

## 2.2 Preparing Dataset

We have created two training dataset based on the features generated from 2.1. In the first dataset, we splitted the original dataset to day and night trajectories and apply features generation on them according to features described in 2.1. So, this operation doubles the number of features, and we call this dataset as *split*. In the second dataset, we consider all trajectories together apply features generation process. We call this dataset as *together*. The test dataset were created on similar ways.

## 3 The Winning Approach

### 3.1 Modeling

We have trained 10 binary predictive models on each of the dataset created in 2.2. All together we trained 20 models with 5 fold cross validations, and predicted on the test dataset. The trained models are:

- Gradient boosted decision trees [1]: Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. We modeled variants of Gradient boosted decision trees:
  - XGBoost [2]: a distributed gradient boosting library designed to be highly efficient, flexible and portable. We modeled two version of XGBoost with two different loss functions:
    1. binary logistic: logistic regression for binary classification, output probability. (model name which we call *xgb\_binary*)
    2. pairwise rank: set XGBoost to do ranking task by minimizing the pairwise loss. (model name which we call *xgb\_rank*)
  - LightGBM [3]: an effective gradient boosting decision tree which contains two novel techniques: Gradient-based One-Side Sampling and Exclusive Feature Bundling to deal with large number of data instances and large number of features respectively. We modeled two version of LightGBM with two different boosting types:
    1. gbd: traditional Gradient Boosting Decision Tree. (model name which we call *lgb\_gbd*)
    2. rf: Random Forest. (model name which we call *lgb\_rf*)
  - CatBoost [4]: a state-of-the-art gradient boosting on decision trees library, which support for both numerical and categorical features. (model name which we call *cat*)
  - GradientBoostingClassifier [1]: a Gradient Boosting for classification algorithm from scikit-learn [5] library. It builds an additive model in a forward stage-wise fashion and it allows for the optimization of arbitrary differentiable loss functions. (model name which we call *sk\_gbt*)
  - RandomForestClassifier [6]: a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset to improve the predictive accuracy and control over-fitting. This algorithm is from scikit-learn [5] library. (model name which we call *sk\_rf*)
  - ExtraTreesClassifier [7]: a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset to improve the predictive accuracy and control over-fitting. This algorithm is from scikit-learn [5] library. (model name which we call *sk\_et*)
  - SVC [8]: a libsvm based Support Vector Machines estimator from scikit-learn [5] library. (model name which we call *sk\_svc*)
  - GPC [9]: a probabilistic predictions with Gaussian process classification estimator from scikit-learn [5] library. (model name which we call *sk\_gpc*)

The model parameters are identified by 5 fold cross validations on each of the algorithms on both *together* and *split* dataset. Thus all the model name will be prefixed with *together* and *split* to distinguish. Models estimated on *together* dataset are called: *together\_xgb\_rank*, *together\_xgb\_binary*, *together\_lgb\_gbd*, *together\_lgb\_rf*, *together\_cat*, *together\_sk\_gbt*, *together\_sk\_rf*, *together\_sk\_et*,

Table 1: 5 fold cross validation F1 score

Model		
Type	Name	F1 score (cross validation)
Base	<i>together_xgb_rank</i>	0.771361
	<i>together_xgb_binary</i>	0.768704
	<i>together_lgb_gbd</i>	0.765479
	<i>together_lgb_rf</i>	0.758018
	<i>together_cat</i>	0.732773
	<i>together_sk_gbt</i>	0.768177
	<i>together_sk_rf</i>	0.761317
	<i>together_sk_et</i>	0.761673
	<i>together_sk_svc</i>	0.725569
	<i>together_sk_gpc</i>	0.717157
	<i>split_xgb_rank</i>	0.767530
	<i>split_xgb_binary</i>	0.767530
	<i>split_lgb_gbd</i>	0.767029
	<i>split_lgb_rf</i>	0.755920
	<i>split_cat</i>	0.734398
	<i>split_sk_gbt</i>	0.765193
	<i>split_sk_rf</i>	0.759317
	<i>split_lgb_et</i>	0.752338
	<i>split_lgb_svc</i>	0.725369
	<i>split_lgb_gpc</i>	0.714567
Ensemble	Ensemble	0.793456

*together\_sk\_svc*, and *together\_sk\_gpc*. Models estimated on *split* dataset are called: *split\_xgb\_rank*, *split\_xgb\_binary*, *split\_lgb\_gbd*, *split\_lgb\_rf*, *split\_cat*, *split\_sk\_gbt*, *split\_sk\_rf*, *split\_sk\_et*, *split\_sk\_svc*, and *split\_sk\_gpc*. The cross validation results are described in the section 4. Finally, we report the simple majority vote ensemble of all predictions generated from 20 trained models.

## 4 Experimental Results

In the competition setup, we needed to submit directly target on the classification decision boundary without producing the probability estimation. This kind of prediction is called hard classification problem, in contrast to soft classification problem where we could submit probability estimation. The evaluation metric is Accuracy which is also hard to optimize if the prediction problem is unbalanced. We decided to train all the models listed in 3.1 to optimize for better F1 score [10]. The F1 score is the harmonic average of the precision and recall, often perform better in classification problem settings. Thus, F1 score was the stopping criterion at the time of finding optimal parameters of each models.

### 4.1 Cross Validation

As the competition testing dataset is small and submission of predicted classification is randomly modified and reported, we can not reliably compare models on leaderboard results. Thus we decided stick on the cross validation F1 score to tune models and select models in the development phase. A 5 fold cross validation [11] strategies was used throughout the development phase. The 5 fold cross validation split is same for all models and for both custom *together* and *split* dataset. Table 1 shows the 5 fold cross validation scores of our various models on our custom dataset as well as on the ensemble ones. The different hyperparameters of each model have been tuned and used in the final submission models in subsection 4.2.

Table 2: Models accuracy on test dataset (Final standing)

Model	Accuracy
Ours (First-place team)	0.7200
Second-place team	0.7018
Third-place team	0.6909
Median competition scores	0.6436
Average competition scores	0.6181

## 4.2 Final Submission

To reduce the effect of random seeds in the 20 models setting, we simulated each model settings for 10 times with 10 different random seeds. For each model setting, we used the hyperparameters found in the previous subsection 4.1 except number of iterations of a particular model, which the model determine dynamically. Finally, we applied simple majority vote ensemble [12] on all 200 hard prediction on the test dataset. Table 2 shows the testing scores from the final phase of the competition <sup>5</sup>.

## 5 Conclusion

We introduced an ensemble learning approach to predict gender of shearwater based on trajectory and associated metadata. The ensemble learning approach was able to model trajectories to output the predictions. One potential limitation of our approach is not considering cluster based information, which might be valuable to increase accuracy. As a future direction, we want to make cluster of grids from the trajectories and analyze the problem. One more interesting feature to see how 3D (latitude, longitude, and elevation) clustering might help increase accuracy. In aligned with this, it would be good to have date of the birds’s trajectories, so that we might benefit from seasonal pattern of water level (elevation).

## References

- [1] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.
- [2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 785–794, New York, NY, USA, 2016. ACM.
- [3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017.
- [4] Anna Veronika Dorogush and Vasily Ershov. Catboost : gradient boosting with categorical features support. In *Workshop on ML Systems at NIPS*, 2017.
- [5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [6] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.
- [7] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, April 2006.
- [8] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.
- [9] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

<sup>5</sup><https://competitions.codalab.org/competitions/16283#results>

- [10] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *Proceedings of the 27th European Conference on Advances in Information Retrieval Research*, ECIR'05, pages 345–359, Berlin, Heidelberg, 2005. Springer-Verlag.
- [11] M. Stone. Cross-validators choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.
- [12] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag.