# Can you *on-sight* it?

**Victor Zhang**
Department of Mechanical Engineering
zhangvwk@stanford.edu

## 1 Introduction

As a climbing enthusiast, one question that crosses my mind as I am about to try out a new climbing route is: can I complete this route in one go with no prior information? In the climbing vernacular, this would be called an *on-sight*. Being able to "on-sight" a route demands a certain mastery of the climbing art – at least with respect to the difficulty and type of said route. In fact, aside from the personal satisfaction of on-sighting, professional climbers actually earn more points in competition when they are able to finish a route in only one go.

The question now is, are there distinctive features that determine the ability of one to on-sight a certain route? Put more formally, given our chosen set of 8 features comprised of user and route specifics (see Section 3), we use a wide array of different models (see Section 4) to classify whether one can on-sight a route or not, using the dataset detailed in Section 3. Section 5 shows the experimental results, followed by a discussion.

## 2 Related work

Several surveys on climbing performance exist in the form of articles intended for the general public [6, 5]. However, to my knowledge very few comprehensive statistical or machine learning studies have been undertaken in the past for a classification problem in the context of rock climbing, let alone to motivate the question of on-sighting a route. [2] is arguably the closest study to this present proposal that is publicly available, though the author decided to restrict his analysis to standard linear regression only, while also striving to answer a different question. [1] is also interesting in itself but differs from our motivation in that the authors strove to classify the difficulty of climbing routes based on visual input, which is a whole other problem (the dataset used is thus far different from ours).

## 3 Dataset and Features

### 3.1 Description

**8a** [1] is the world's largest rock climbing logbook, used to track completed rock climbs all over the planet. A user that completes a climb puts in additional information including whether it was *on-sighted* or otherwise. The climb itself includes useful information such as its consensus difficulty grade and rating, location, and type (*lead* – tall route climbed with a rope, or *boulder* – short route climbed without a rope). Finally, user details can optionally include personal physical specifications such as gender, height, weight, age and climbing experience (in terms of duration). The users and ascents' information from this logbook were scraped and collected on 2017-9-13 as per the data description. The dataset is posted publicly on Kaggle [2]. The raw data contains the following four dataframes: *users* ($\approx$ 63k rows, 22 columns), *ascents* ($\approx$ 4.1 million rows, 28 columns), *climbing methods* (5 rows, 4 columns), and *climbing grades* (83 rows, 14 columns).

### 3.2 Preprocessing

The raw dataset included features that are arguably irrelevant to the task at hand. In the `users` dataframe for example, information such as first and last names, interests and occupation (among others) was deliberately left out especially considering it wasn't

[1] https://www.8a.nu/

[2] https://www.kaggle.com/dcohen21/8anu-climbing-logbook

CS229: Machine Learning, Spring 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

in a standardized format. Similarly, several irrelevant and non-standardized features from the `ascents`, `methods` and `grades` dataframes were removed, such that the only remaining features are:

```
country, is_female, height, started, birth, grade_id, is_bouldering, year
```

The first five features are user features, where `started` denotes the year in which the user started climbing. The latter three are features specific to an ascent, where `year` denotes the year in which the user performed the climb. The goal is thus to predict the binary label `is_onsight` given the above features.

Additional preprocessing was applied to filter out bogus values. Furthermore, since `height` and `weight` were found to be correlated, it was decided to drop the weight feature as height is generally a more determining factor in climbing performance among seasoned climbers (which is what this dataset is made of).

As a significant portion of the dataset had missing `height`, `weight` and/or `started` data, a multivariate imputer was used to estimate the missing data from other features. Note that this is to populate a larger training set only, while our model is then evaluated on a test set made of a selection of complete data from the original and un-imputed dataset. Formally, let $\mathcal{S}_{\text{incomplete}}$ denote the original (incomplete) dataset. From $\mathcal{S}_{\text{incomplete}}$, we select a complete dataset $\mathcal{S}_{\text{complete}} \subset \mathcal{S}_{\text{incomplete}}$. We then split $\mathcal{S}_{\text{complete}}$ into a training set $\mathcal{S}_{\text{train, orig}}$, a validation set $\mathcal{S}_{\text{dev}}$, and a test set $\mathcal{S}_{\text{test}}$ with a ratio of 70/20/10. Let $\mathcal{S}_{\text{imputed}}$ be the dataset after imputing the data in $\mathcal{S}_{\text{incomplete}}$, and let $\mathcal{S}_{\text{train, imp}} = \mathcal{S}_{\text{imputed}} \setminus \{\mathcal{S}_{\text{test}} \cup \mathcal{S}_{\text{dev}}\}$. $\mathcal{S}_{\text{train, imp}}$ thus constitutes an alternate, significantly larger training set than $\mathcal{S}_{\text{train, orig}}$.

The goal is to maximize accuracy on $\mathcal{S}_{\text{dev}}$ and then to evaluate the final fine-tuned models on $\mathcal{S}_{\text{test}}$. We experiment with the two different training sets $\mathcal{S}_{\text{train, orig}}$ and $\mathcal{S}_{\text{train, imp}}$, hopefully observing better results with the latter.

### 3.3 Preliminary analysis

An overview of the statistics of the two training sets $\mathcal{S}_{\text{train, orig}}$, $\mathcal{S}_{\text{train, imp}}$ and the test set $\mathcal{S}_{\text{test}}$ is given in Table 1.

| | Set size ($10^6$) | `is_onsight` (%) | `is_bouldering` (%) | `is_female` (%) |
|---|---|---|---|---|
| $\mathcal{S}_{\text{train,orig}}$ | $\approx 1.76$ | 32.02 | 29.41 | 7.92 |
| $\mathcal{S}_{\text{train,imp}}$ | $\approx 2.17$ | 31.81 | 28.98 | 10.50 |
| $\mathcal{S}_{\text{test}}$ | $\approx 0.433$ | 30.14 | 30.84 | 9.36 |

Table 1: Comparison table of $\mathcal{S}_{\text{train, orig}}$, $\mathcal{S}_{\text{train, imp}}$ and $\mathcal{S}_{\text{test}}$ on the label `is_onsight` and the two binary-valued features `is_bouldering` and `is_female`.

Additionally, histogram similarities (in the sense of the intersection over 20 bins) for the features `birth`, `height` and `grade_id` are shown in Figure 1. We observe that though the statistics are all similar (with the imputed dataset seemingly better representing the test set distribution on the label and the female proportion), there is skew towards non-onsighted, non-bouldering (i.e. lead-climbing) and non-female (i.e. male) completed ascents. Refer to Section 4 for the methods used to remedy this imbalance.

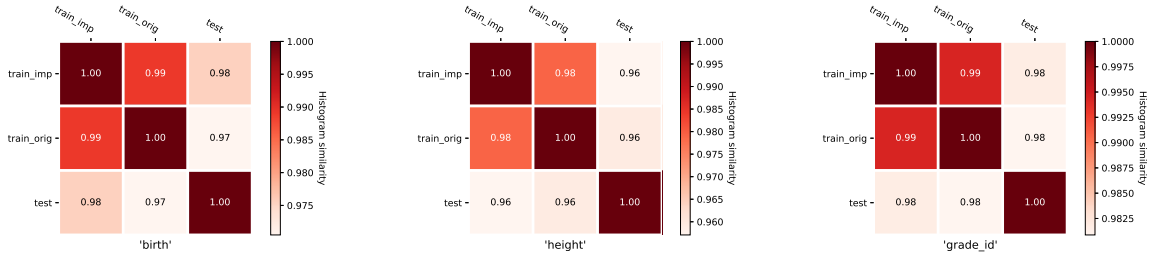


Figure 1: Histogram similarity for the features `birth`, `height` and `grade_id` from top left to bottom right, respectively.

## 4 Methods

### 4.1 Models

For this classification task, we attempt the following approaches: logistic regression, support vector machines (SVMs), multi-layer perceptron (MLP) as well as ensemble methods. For each approach, different hyperparameters and configurations were carried out. A brief review of these methods is given in this section.

Let $\{(x^{(i)}, y^{(i)}) \mid i \in [\![1, n]\!]\}$ be a sequence of $n$ independent draws from a joint distribution $P_{XY}$. The variable $X$ takes values in $\mathbb{R}^d$ with $d \in \mathbb{N}^*$ the number of features and the variable $Y$ (also called the label) takes values in $\{0, 1\}$. Let $\epsilon(\hat{h}_n) > 0$ denote the excess risk $\mathbb{P}(Y \neq \hat{h}_n(X)) - \mathbb{P}(Y \neq h^*(X))$, where $h^*$ is the optimal Bayes classifier which is defined as follows:

$$h^*(X) = \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1|X) > 1/2 \\ 0 & \text{if } \mathbb{P}(Y = 1|X) \leq 1/2 \end{cases}$$

The goal is to build a classifier $\hat{h}_n(X)$ such that the excess risk $\epsilon(\hat{h}_n)$ is as small as possible.

- Logistic regression consists in maximizing the log-likelihood of the data assuming $\eta(X) := \mathbb{P}(Y = 1|X; \theta) = 1/(1 + e^{-\theta^\top X})$. The loss to be minimized is

$$\ell(\theta) = -\sum_{i=1}^{n} y^{(i)} \left[\log \eta(x^{(i)}) + (1 - y^{(i)}) \log(1 - \eta(x^{(i)}))\right] + \lambda \|\theta\|_p^p$$

  where the rightmost contribution is a ($\ell_p$ for $p \in \{1, 2\}$) regularization term and serves the purpose of decreasing the variance of the model.

- Linear support vector classification solves the following convex optimization problem:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{2} \|\theta\|_2^2 + C \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)} \left(\theta^\top x^{(i)}\right) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \quad i = 1, \ldots, m \end{aligned}$$

  which is the optimal margin classifier after $\ell_1$ regularization (a necessary term for non-linearly separable data which is the case here as observed later on). Though one can map the input to a higher-dimensional feature space and efficiently solve this problem with kernels, linear SVM turned out to suffice as shown in Section 5. It is to be noted that the training complexity is quadratic in the number of samples which makes SVM a method that does not scale very well to our large training sets (see Section 3.3). In practice, it was decided, for the sake of computation time, that only a random subset of length $n_{\text{samples}}$ would be drawn from the training set.

- Multi-layer perceptron is a fully connected network with an arbitrary number of layers which can each be of arbitrary size. The activation functions are denoted $a$, and the function preceding the output is the sigmoid (as for logistic regression) for binary classification. A regularization term can be added to the loss as well to prevent overfitting the training set.

- Ensemble methods employed in this study are (i) random forests and (ii) AdaBoost.
  (i) Random forests [3] is a bagging method which consists in drawing with replacement $B$ random samples $(X_b, Y_b)$ from the dataset, fitting a regression tree $f_b$ on $(X_b, Y_b)$ while selecting, at each candidate split in the learning process, a random subset of the features to decorrelate the trees. The classifier is then the average of the $f_b$'s, for $b \in [\![1, B]\!]$.
  (ii) AdaBoost [7] is a boosting method which consists in building the classifier on $B$ weak learners sequentially (as opposed to in parallel in the bagging approach) so as to assign a weight to each of them, while using a shrinking parameter $\alpha$ to slow down learning and prevent overfitting. The final classifier is then the weighted average of the weak learners. AdaBoost is a particular boosting method in which incorrect classification receives higher weights, prompting its classifiers to pay more attention to them in the next iteration, and vice versa.

### 4.2 Metrics

Since the datasets (both training and test) are unbalanced (from Table 1, around 70 % of the points are classified as 0), we expect poor precision and recall, and a high specificity. For training, the two following options are: either use the unbalanced data as is and use modeling methods to combat the imbalance, or simply use random undersampling such that each class represents half of the dataset. While the former option was contemplated, since undersampling would not greatly decrease the training set size relative to the test set and thus not significantly alter the outcome, the latter option was chosen. Experimental results indeed showed that the latter was the way to go (see Section 5.1).

Since the test set is heavily skewed, the metric chosen is the F1 score. Note that, had the test set not been skewed (e.g. deliberately balanced as with the training set), we would have used the AUROC (Area Under the Receiver Operating Characteristic) to measure the performance of a model because it would have better characterized the performance in identifying each class equally. The test set should be an estimate of the real-life statistics, and since it seems reasonable to assume these are actually skewed (at least with respect to the proportion of on-sighted and female-completed routes), it was decided to keep the test set unbalanced and consequently evaluate the models using the F1 score.

## 5 Experiments and discussion

### 5.1 Unbalanced versus balanced

One possible way to deal with an unbalanced dataset is to use boosting. Indeed, the method is used to generate several training datasets with weights that tip the scales in favor of the most difficult cases, i.e. the minority class. One such method is AdaBoost [7], which we ran on $\mathcal{S}_{\text{train, orig}}$ and $\mathcal{S}_{\text{bal, orig}}$. The former was described above in Section 3.2, while the latter was obtained using random undersampling of the former such that each class is equally represented. The algorithm was run using 50 decision tree classifiers with a maximum depth of 1 as base estimators. The results are shown on Table 2, after evaluating the classifiers on the test set.

| | $\mathcal{S}_{\text{train,orig}}$ (%) | $\mathcal{S}_{\text{bal,orig}}$ (%) |
|---|---|---|
| Precision | 60.22 | 51.11 |
| Recall | 51.18 | 78.04 |
| F1 score | 55.33 | 61.77 |

Table 2: Comparison table of results obtained on the test set after training AdaBoost on $\mathcal{S}_{\text{train, orig}}$ and $\mathcal{S}_{\text{bal, orig}}$.

We observe that using the balanced dataset for training increases the F1-score by approximately 12%. Therefore, coming back to the point made previously in Section 4.2, it was decided to deal with the skewed statistics by simply undersampling the training set.

### 5.2 Original versus imputed

As detailed in Section 3.2, an alternative training dataset, $\mathcal{S}_{\text{train,imp}}$ was generated using additional imputed values. As shown in the previous subsection, since we are dealing with imbalance using undersampling, another balanced dataset was created out of $\mathcal{S}_{\text{train,imp}}$, which we denote $\mathcal{S}_{\text{bal,imp}}$. The hope was that using this larger training set would result in a better generalization and consequently better test results. AdaBoost with the same setup was run on the two balanced training sets $\mathcal{S}_{\text{bal,orig}}$ and $\mathcal{S}_{\text{bal,imp}}$, with the results shown in Table 3, after evaluating the classifiers on the test set.

| | $\mathcal{S}_{\text{bal,orig}}$ (%) | $\mathcal{S}_{\text{bal,imp}}$ (%) |
|---|---|---|
| Precision | 51.11 | 50.30 |
| Recall | 78.04 | 83.66 |
| F1 score | 61.77 | 62.82 |

Table 3: Comparison table of results obtained on the test set after training AdaBoost on $\mathcal{S}_{\text{bal, orig}}$ and $\mathcal{S}_{\text{bal, imp}}$.

We observe a slight increase in the F1 score when using the model trained on the larger balanced dataset $\mathcal{S}_{\text{bal,imp}}$. Thus it was decided to move forward with $\mathcal{S}_{\text{bal,imp}}$ and drop $\mathcal{S}_{\text{bal,orig}}$. Note that we are now using a dataset of $|\mathcal{S}_{\text{bal,imp}}| \approx 1.38 \cdot 10^6$ points.

### 5.3 Comparisons and discussion

A comparison of the results obtained with 5 different models is shown in Table 4, after evaluating the classifiers on the test set. The corresponding hyperparameters for each model are shown in Table 5.

| | Logistic regression (%) | SVM (%) | Random forests (%) | AdaBoost (%) | MLP (%) |
|---|---|---|---|---|---|
| Training accuracy | 65.64 | 75.19 | 73.57 | 74.87 | 75.16 |
| Training F1 score | 71.06 | 75.95 | 75.25 | 76.23 | 77.15 |
| Precision | 40.69 | 49.03 | 49.12 | 50.30 | 49.13 |
| Recall | 91.55 | 87.34 | 83.04 | 83.66 | 88.20 |
| F1 score | 56.34 | 62.80 | 61.73 | 62.82 | 63.11 |

Table 4: Model comparison of results obtained on the test set after training on $\mathcal{S}_{\text{bal, imp}}$.

The main takeaway observation is that logistic regression, the “simplest" linear classifier, achieves an F1 score that is not that much lower to that of a 4-layer fully connected network. In other words, it seems challenging to find suitable non-linearities that would greatly outperform a simple linear model. Figure 2 shows a subset of the training data projected onto the first two

| | Hyperparameters |
|---|---|
| Logistic regression | *Regularization*: $\ell_2$, $\lambda = 1$ |
| SVM | *Kernel:* linear<br>$C = 1$<br>$n_{\text{samples}} = 10^4$ |
| Random forests | $B = 100$ |
| AdaBoost | $B = 100$<br>$\alpha = 1$<br>max_depth = 1 |
| MLP | *Layer dimensions:* $(5, 5, 5, 5)$<br>$a$: ReLU<br>*Regularization:* $\ell_1$, $\lambda = 1$<br>*Optimization:* Adam (initial l.r. $10^{-3}$)<br>*Batch-size:* 200 |

Table 5: Hyperparameters corresponding to the models exposed in Table 4. A tolerance of $10^{-4}$ was set for any implementation requiring one.

principal eigenvectors of the training set $\mathcal{S}_{\text{bal,imp}}$, and it indeed seems non-trivial to separate the two classes even in the first two principal dimensions: the two classes are mixed together with no obvious coherence – linear or otherwise. It therefore comes to no great surprise that the non-linear classifiers have trouble finding the relevant weights that would result in a much better performance than the linear ones – in fact, the training accuracy of each row shows that the models aren't capable of overfitting past around 75%.

Despite the poor performance of the advanced non-linear methods relatively to the logistic regression baseline, this study was nonetheless useful to practically confirm the following: decorrelating the trees with random forests does increase generalization, AdaBoost does not need its base estimators to be complex (the maximum depth of each decision tree `max_depth` is only 1) in order to produce encouraging results, and neural networks perform better when the structure is deep (e.g. (5,5,5,5)) rather than wide (e.g. (50,50)), thereby learning features at various levels of abstraction while not overfitting the training set.

## 6 Conclusion and future work

We achieved a baseline F1 score of around 56% using regularized logistic regression and struggled to increase it by a great amount, with the most successful method being a 4-layer FC network achieving around 63% (thus making for around a 12 % increase). Future work would include testing other methods such as kernel-induced random forests [8] and boosting with more advanced base learners (e.g. an FC network), though it would be surprising to see any substantial increase in performance given our current results. A more sound way of going about this issue would perhaps be to use Fisher discriminant analysis to infer features that maximize the class separability, include additional features to render the data more "learnable", and to try out other reweighting schemes such as class-balanced loss [4].

In addition, this dataset in itself comes from seasoned climbers and not the general public; as such, the proportions seen in Section 3.3 are probably not representative of, and the predictions do not generalize to the general climbing population.

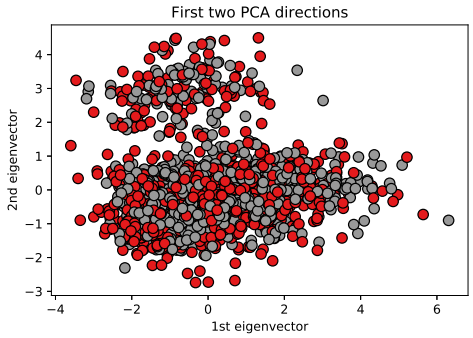


Figure 2: A subset of the training set $\mathcal{S}_{\text{bal, imp}}$ projected on the first two principal eigenvectors.

## References


[1] Juan Carlos Sarmiento Alejandro Dobles and Peter Satterthwaite. Machine learning methods for climbing route classification. http://cs229.stanford.edu/proj2017/final-reports/5232206.pdf, 2017.

[2] Steve Bachmeier. Rock climber characteristic analysis or: Does being tall really help? https://github.com/stevebachmeier/climber-characteristic-analysis, 2019.

[3] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[4] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples. *arXiv e-prints*, page arXiv:1901.05555, Jan 2019.

[5] Naftali Harris. A statistical analysis of climbing. https://www.naftaliharris.com/blog/climbing-statistical-analysis/, 2014.

[6] Chris Ring. The height of injustice: Is being tall an advantage in your climbing career? https://rockandice.com/climbing-news/the-height-of-injustice-is-being-tall-an-advantage-in-your-climbing-career, 2018.

[7] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[8] Erwan Scornet. Random forests and kernel methods. *IEEE Trans. Inf. Theor.*, 62(3):1485–1500, March 2016.