Classification trees, random forests

Marko Medved

I. Part 1: Implementing multinomial logistic regression and ordinal logistic regression

A. Implementation details

The first model implemented is multinomial logistic regression. In this approach, we calculate linear predictors for each class, excluding the reference class. Afterward, we apply the softmax function to compute the probability for the true class. To optimize the model's weights, we use the maximum likelihood estimation (MLE) method, leveraging the *fmin_l_bfgs_b* with numerical gradients.

An important aspect of this implementation is the inclusion of an intercept column. This is necessary because, in some cases, all feature values may be zero. Without the intercept, this would lead to zero probabilities for all classes. By including the intercept, we ensure that the model can still generate non-zero probabilities for all classes, even when the features are zero.

During the prediction phase, we apply the softmax function to the linear predictors for each class at each data point. This gives us the predicted probabilities for all classes, which are then reported as the model's predictions.

Next, ordinal logistic regression was implemented. The main difference with multinomial logistic regression is the assumption that the target categories have a natural ordering. This assumption allows us to significantly reduce the number of parameters. Instead of calculating separate predictors for each class, we use just one linear predictor along with thresholds (which are fewer than the number of classes) to determine the boundaries between classes.

During the model building phase, we again use the MLE approach, as we did with multinomial logistic regression, and the same optimizer. However, instead of explicitly modeling the thresholds, we model the differences between the thresholds. To ensure numerical stability, we constrain these differences (the deltas) to be greater than 1 during the optimization process.

This reduction in parameters, combined with the ordered nature of the categories, makes ordinal logistic regression more efficient for datasets with ordered categorical targets.

B. Testing our implementation on the basketball dataset

We evaluated both classifiers on the provided dataset to predict shot type based on various features. To preprocess the data, we One-Hot encoded the categorical variables and dropped one category from each feature to avoid multicollinearity. Furthermore, we identified strong correlations between certain features. Specifically, the "TwoLegged" and "No Movement" features, as well as the player type indicators "Guard" and "Forward," exhibited high correlation, which we can observe on fiure. As a result, we removed the "TwoLegged" and "Forward" features to prevent convergence problems and reduce redundancy.

The model evaluation was conducted using 10-fold cross-validation, and the uncertainty in the results was calculated under the assumption of asymptotic normality. The results are presented in Table I. As expected the multinomial version outperformed the ordinal one, since there is no clear natural order for shot types in basketball.

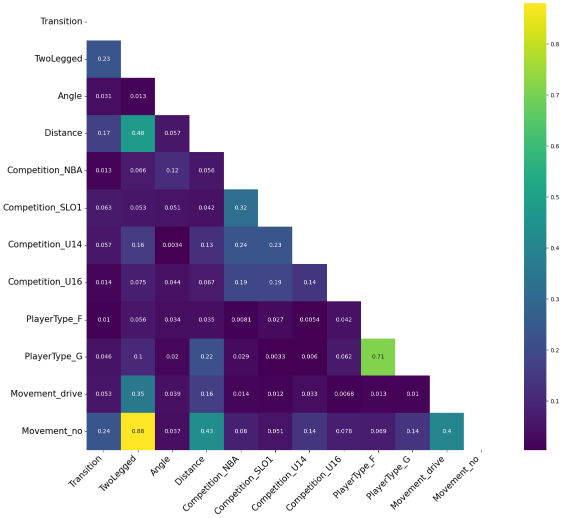

Figure 1. Absolute correlation of features in the basketball dataset.

Model	Accuracy	Log loss
Multinomial	0.7353 ± 0.0062	0.667 ± 0.012
Ordinal	0.7138 ± 0.0064	0.985 ± 0.018

Table I
Comparison of results using multinomial and ordinal logistic regression

II. part 2.1: Application of the Multinomial regression

In this analysis, we explored the relationships between shot type and various features. To gain insight, we directly compared the coefficients from the linear predictors. These coefficients represent how the log-odds of a given shot type change when a particular feature is adjusted, while holding other features constant.

To estimate the uncertainty of these coefficients, we employed bootstrap sampling. We repeatedly fitted the model to bootstrap samples, calculated the mean of the betas as our best estimate, and used the standard deviation as the measure of uncertainty. We used the "tip-in" shot type as our reference class.

The results of our analysis are displayed in Figure 2. From the plot, we observe that some features exhibit high uncertainty compared to the value of coefficients in certain shot types, suggesting that these features may not contribute significantly to differentiating those shot types from the reference class. On the other hand, several features show consistently high coefficients across multiple shot types.

For instance, the "Movement_no" coefficient is notably influential. The likelihood of no movement would intuitevely be high for the "above head" shot. However, this comparison is made against the "tip-in" shot, where the only movement in the dataset is "Movement_no"—which is also true for "dunk" and "hook shot" categories. In contrast, "Movement_drive" has a clear, intuitive impact. For example, a drive is less likely to be associated with an "above head" shot but is much more common for "layup" and "other" shots. For the "tip-in", "hook shot", and "dunk", the likelihood of a drive is relatively similar to the reference class, as these shots typically require less dynamic movement.

The "Distance" feature also shows an interesting pattern: the "tip-in" shot is associated with the shortest distances, followed by the "dunk", "layup", and "hook shot". Both "other" and "above head" shots, on the other hand, have higher coefficients, which aligns with their nature (mostly for the above head), as they often occur from farther distances.

When examining the "Competition_U14" feature, we see that "tip-in" is less likely compared to all shot types except for the "dunk". This makes sense, as younger players (under 14) are unlikely to perform tip-ins and even less likely to dunk.

Finally, the intercept plays a significant role across all shot types. It represents the baseline log-odds of selecting a particular shot type when all other features are zero.

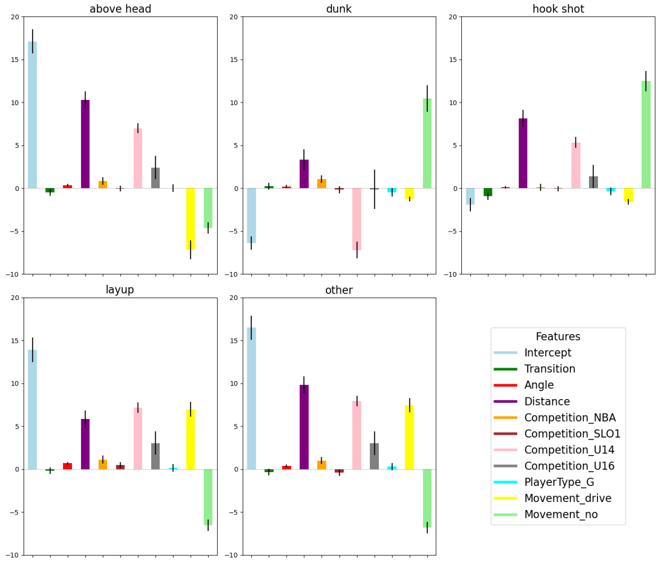

Figure 2. Comparison of relationships of features and different shot types

III. part 2.2: Application of the Ordinal regression

In this section, we designed a data-generating process (DGP) where ordinal logistic regression outperforms multinomial logistic regression. The process was designed as follows: We begin by assigning a class to each data point. For each class, we generate the corresponding feature values by sampling from a uniform distribution spanning from i - 0.5 to i + 0.5, where i is the current class. This ensures that the data is ordinal in nature. To introduce some variability and reduce the correlation between features, we then add noise sampled from a standard normal distribution.

We tested the DGP using a dataset with 100 training data points and 100 testing data points. The results of this experiment can be seen in Table II.

IV. Part 3: GLM Diagnostics

In this section, we first implemented linear regression under the assumption that the residuals follow a normal distribution, using the least squares method. We applied this model to predict the basketball shot distance based on the shot angle.

Model	**Accuracy**	**Log Loss**
Multinomial Logistic Regression	0.580 ± 0.050	0.942 ± 0.100
Ordinal Logistic Regression	0.630 ± 0.049	0.823 ± 0.070

Table II

Comparison of accuracy and log loss between multinomial and ordinal logistic regression models.

As part of our diagnostic checks, we plotted the Q-Q plot (Figure 3) to assess whether the assumption of normally distributed residuals holds. In the Q-Q plot, we compare the quantile residuals to the theoretical quantiles of a standard normal distribution. If the residuals were normally distributed, the points would align closely with the reference line. However, as shown in the plot, the residuals deviate significantly from this line, indicating that the normality assumption is not justified.

Notably, the residuals exhibit a bi-modal distribution, which is consistent with the distribution of the shot distances themselves, as shown in Figure 4. This is not suprising since the distance and angle do not have much correlation, so we would have a similar distribution of data at each angle.

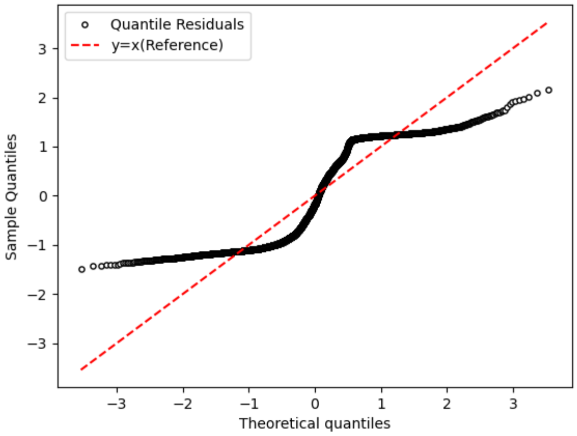

Figure 3. Quantile-quantile plot

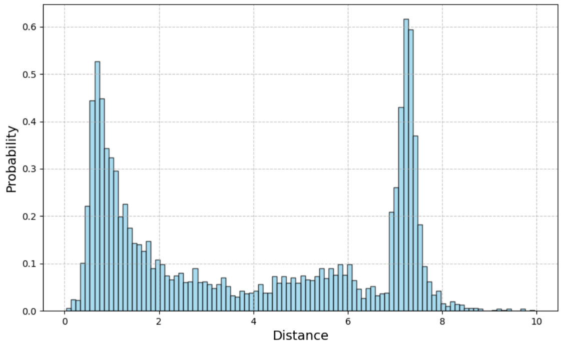

Figure 4. Distribution of the distance of basketball shots

The second plot is the quantile residuals vs fitted values plot. We chose to use quantile residuals because they tend to have approximately constant variance, which makes them suitable for assessing homoscedasticity (the assumption of constant variance of residuals). This plot helps us verify whether the variance of the residuals changes across the range of fitted values.

In Figure 5, we observe that there is no apparent trend or pattern in the variance of the residuals as the fitted values increase. This suggests that the residuals are homoscedastic.

As observed before, the distribution of the residuals remains bi-modal, a characteristic also observed in the original distribution of the basketball shot distances (Figure 4).

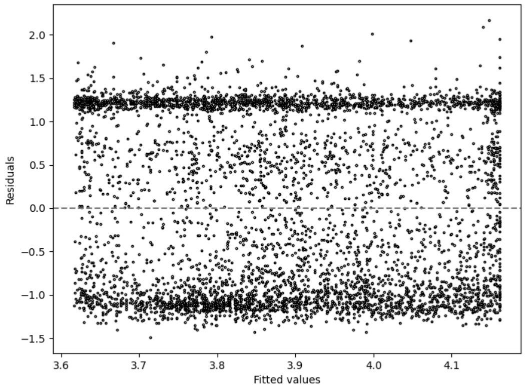

Figure 5. Residuals vs fitted values

The last plot, shown in Figure 6, represents Cook's distance for each data point. Cook's distance measures the influence of individual data points on the regression model, taking into account both the features and the target values. A high Cook's distance indicates that a data point has a significant impact on the model, potentially altering the estimated regression coefficients.

In the plot, we observe that none of the points have a particularly high Cook's distance. As a rule of thumb, data points with a Cook's distance close to than 1 are considered highly influential. Since all the points lie way below this threshold, we can conclude that there are no particularly influential data points that would unduly affect the model. Therefore, it is safe to assume that the regression results are not being significantly impacted by any single data point.

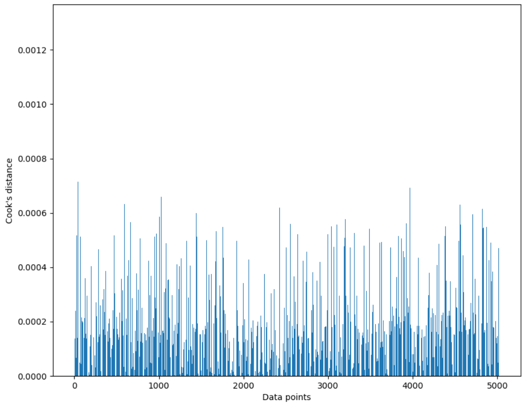

Figure 6. Cook's distance across data points