*Jurnal Ilmiah Matematika dan Terapan*
Volume 23 Number 1 June 2026 (Pages 54 - 63)
doi: https://doi.org/10.22487/2540766X.2026.v23.i1.18187

JIMT
Jurnal Ilmiah Matematika dan Terapan
ISSN : 2540 – 766X

# BBQ WEATHER PREDICTION IN BASEL USING ENSEMBLE MACHINE LEARNING

**Royyan Amigo[1], Reyhan Ksatria Brahmacarya[2], Muhamad Hilman Rizaldi[3]**

[1]Universitas Pesantren Tinggi Darul Ulum
[2,3]Universitas Brawijaya
[1]royyanamigo@saintek.unipdu.ac.id, [2]reyhanksatria05@student.ub.ac.id, [3]hilmanrizaldi26@gmail.com

**ABSTRACT**

Weather-dependent decision making, such as planning an outdoor barbecue (BBQ), benefits from accurate short-term weather forecasts. This study addresses label leakage in weather classification by using next-day forecasting targets derived from same-day observations. To account for class imbalance, a balanced evaluation framework was adopted, with F1-score as the primary metric. Five machine learning classifiers were developed and evaluated using feature weighting and feature selection techniques under chronological training-validation-test splits. Performance differences were further assessed using bootstrap resampling and McNemar's test. Among the evaluated models, RUSBoost combined with weighted voting and feature selection achieved the best overall performance, with an F1-score of 0.6841, ROC-AUC of 0.9076, and PR-AUC of 0.6961 on the selected feature subset. The selected feature subset was statistically indistinguishable from the full feature set (McNemar's test, $p = 0.8388$), whereas the proposed ensemble significantly outperformed the baseline classifier (McNemar's test, $p = 0.032$). These findings demonstrate that an ensemble-based approach with appropriate feature selection provides accurate and reliable next-day weather classification while improving model interpretability and supporting trustworthy decision making.

**Keywords : Class Imbalance, Ensemble Learning, Feature Selection, McNemar Test, Weather Forecasting**

# I. INTRODUCTION

Weather-dependent recreational planning, such as deciding whether the following day is suitable for an outdoor barbecue, represents a small but illustrative example of short-horizon meteorological classification problems. Unlike numerical weather prediction models, which simulate atmospheric processes by solving physical equations, statistical and machine learning approaches learn empirical relationships between observed variables and target conditions directly from historical data [1]. Ensemble methods, which combine multiple heterogeneous base learners, have consistently outperformed single classifiers on tabular meteorological data because different algorithms capture complementary aspects of nonlinearity, feature interactions, and noise in the data [2].

Two methodological risks are frequently overlooked in applied weather classification studies. The first is label leakage, in which the target variable is partially or fully determined by features measured at the same time point, leading to artificially high classification performance that does not reflect genuine predictive ability [3]. The second is validation-set overfitting, which occurs when the same validation set is repeatedly used for model selection, hyperparameter tuning, ensemble weighting, and feature selection. Each additional decision based on the same validation data increases the risk of overly optimistic performance estimates that may not generalize to unseen data [4].

This study uses daily weather records from the European Climate Assessment and Dataset (ECA&D) for Basel, Switzerland, covering the period from 2000 to early 2010 [5]. A binary label indicating whether weather conditions were suitable for an outdoor barbecue was derived from these observations. During preliminary data exploration, the original same-day label was found to be a deterministic function of same-day precipitation, resulting in trivially perfect classifiers with no practical forecasting value. To eliminate this source of label leakage, the problem was reformulated as a next-day prediction task using one-day lagged features, thereby transforming it into a genuine short-horizon forecasting problem with inherent uncertainty.

This study has three objectives. First, it compares five algorithmically diverse classifiers under a rigorous evaluation framework designed for imbalanced classification. Second, it develops and statistically evaluates an ensemble model using multiple combination strategies rather than relying on a single default approach. Third, through a chronological training-validation-test split and formal statistical hypothesis testing (bootstrap resampling and McNemar's test), it demonstrates that feature selection and model selection decisions can be validated with the same rigor as confirmatory model evaluation rather than being justified solely by validation performance.

# II. METHODS

## 2.1. Data Source and Variables

The dataset used in this study is obtained from the Weather Prediction Dataset, provided by the European Climate Assessment and Dataset (ECA&D) project [5], and restricted to observations from Basel, Switzerland. The dataset comprises 3,654 daily observations from 1 January 2000 to 1

January 2010, containing nine continuous meteorological variables: cloud cover, humidity, sea-level pressure, global radiation, precipitation, sunshine duration, mean temperature, minimum temperature, and maximum temperature, together with a binary label indicating whether the weather was suitable for an outdoor barbecue (BBQ_weather). A data quality assessment confirmed that the dataset contained neither duplicate records nor missing values across all columns.

### 2.2. Target Reformulation and Feature Engineering

Exploratory analysis revealed that BBQ_weather = True occurred if and only if same-day precipitation was zero, indicating the original label to be a deterministic, rule-based function rather than a genuine forecasting target. To eliminate this source of label leakage, the target was redefined as the BBQ-suitability label of the following day (target_next_day), and one-day lagged versions of all nine weather variables were constructed as predictors, together with the same-day BBQ_weather value as a persistence feature. An additional engineered feature, temperature range (maximum minus minimum temperature, computed for both the current day and its one-day lag), was retained after it was found to exhibit a stronger correlation with the next-day target ($r = 0.416$) than either temperature variable individually. This process resulted in an initial feature set of 22 variables.

### 2.3. Data Splitting

Because daily temperature exhibited strong temporal autocorrelation (lag-1 autocorrelation of 0.957) and a clearly repeating seasonal cycle, the dataset was partitioned chronologically rather than randomly to prevent information leakage between temporally adjacent observations and to better reflect real-world forecasting conditions. The partition used training data from 2000–2005 (2,191 observations, 59.99%), validation data from 2006–2007 (730 observations, 19.99%), and test data from 2008–2009 (731 observations, 20.02%). The test set was withheld from all subsequent model selection, tuning, ensembling, and feature-selection procedures and was used only once for the final performance evaluation.

### 2.4. Candidate Models and Class Imbalance Handling

Five classifiers with distinct inductive biases were compared: CatBoost and LightGBM (gradient boosting), RUSBoost (boosting with built-in random undersampling for imbalanced data), Nearest Centroid (distance-based), and SGDClassifier (linear, wrapped in a Pipeline with StandardScaler due to its scale sensitivity). A DummyClassifier using the most-frequent strategy served as a naive baseline. Because the target is imbalanced (approximately 74% negative and 26% positive), class-weighting strategies were tested individually for each model against their unweighted defaults. For CatBoost, the positive class weight was adjusted using:

$$\text{scale\_pos\_weight} = \frac{n_{neg}}{n_{pos}} \quad (1)$$

where $n_{\text{neg}}$ is the number of observations belonging to the negative class and $n_{\text{pos}}$ is the number of observations belonging to the positive class.

For LightGBM and SGDClassifier, the balanced class weight was computed for each class $j$ as

$$w_j = \frac{n_{samples}}{K \times n_j} \quad (2)$$

where $n_{samples}$ is the total number of training observations, $K$ is the number of classes, and $n_j$ is the number of observations in class $j$.

RUSBoost, which already performs random undersampling at every boosting iteration, was evaluated with and without additional sample weighting. The default configuration was retained because additional weighting consistently degraded performance, indicating that it was redundant with the built-in resampling mechanism.

### 2.5. Hyperparameter Tuning

Each of the five models was tuned using a Genetic Algorithm (GASearchCV). The search space for each model was designed according to its observed overfitting tendency in the baseline comparison. For models exhibiting a large training-validation F1 gap, the search space emphasized regularization-related hyperparameters (e.g., depth, iterations, and L2 regularization for CatBoost; max_depth, min_child_samples, and reg_lambda for LightGBM). Models with smaller training-validation gaps were tuned over narrower search spaces.

### 2.6. Ensemble Construction

All five tuned models were compared as single classifiers, then combined in every possible pairing, triple, quadruple, and full five-way combination using simple probability averaging (31 combinations in total), thereby establishing a naive baseline for ensembling. Three dedicated ensemble strategies were then applied to the four models supporting predict_proba (all except Nearest Centroid). Weighted Voting searched for an optimal weight vector using 2,000 random draws from a Dirichlet distribution, whose probability density function is

$$f(x_1, \ldots, x_k;\ \alpha_1, \ldots, \alpha_k) = \frac{1}{B(\alpha)} \times \prod_{i=1}^{k} x_i^{\alpha_i - 1} \quad (3)$$

where the weight vector $\mathbf{x} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_K})$ satisfies $x_i \geq 0$ and $\sum x_i = 1$ by construction, $K = 4$ is the number of candidate models, and $B(\alpha)$ is the multivariate Beta normalizing function,

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)} \quad (4)$$

with $\Gamma(\cdot)$ denoting the Gamma function. Symmetric concentration parameters ($\alpha_i = 1$ for all $i$) were used, producing a uniform distribution over the weight simplex and avoiding prior bias toward either balanced or extreme weight combinations. This approach was preferred to independently sampling uniform weights followed by renormalization because, according to the Central Limit Theorem, the latter concentrates samples near the simplex centroid and rarely explores extreme single-model-dominant weight combinations. Stacking used out-of-fold predictions from a 10-fold chronological split as input features to a Logistic Regression meta-learner, with rows that were never assigned to a validation fold discarded. Greedy Ensemble Selection began with the best single model and iteratively

added the model that produced the greatest improvement in F1-score until no further improvement was achieved.

### 2.7. Model Evaluation and Statistical Testing

Because of class imbalance, the F1-score was adopted as the primary evaluation metric rather than accuracy, since a trivial always-negative classifier already attains approximately 74% accuracy while providing zero predictive value for the minority class. F1-score is the harmonic mean of precision $(P)$ and recall $(R)$:

$$F1 = 2 \times \frac{(P \times R)}{(P + R)} \tag{5}$$

where precision and recall are defined as

$$P = \frac{TP}{TP+FP}$$

$$R = \frac{TP}{TP+FN}$$

where $TP$ (True Positive) is the number of correctly predicted positive instances, $FP$ (False Positive) is the number of negative instances incorrectly predicted as positive, and $FN$ (False Negative) is the number of positive instances incorrectly predicted as negative.

Model comparisons were additionally verified using bootstrap resampling (1,000–2,000 iterations) to estimate win rates. When the bootstrap results were inconclusive**,** McNemar's test was applied as a formal paired-classifier significance test. Given a 2×2 contingency table summarizing the agreement and disagreement between two classifiers on the same validation samples, with $b$ and $c$ denoting the discordant cell counts, the McNemar chi-squared statistic is

$$\chi^2 = \frac{(b - c)^2}{(b + c)} \tag{6}$$

evaluated against a chi-squared distribution with one degree of freedom (or the exact binomial form for small samples) to determine whether the observed performance difference is statistically significant [6]. Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves, together with their area-under-curve values (ROC-AUC and PR-AUC), were used as complementary evaluation criteria, given that PR-AUC is generally more informative than ROC-AUC for imbalanced datasets because it is less affected by the large number of true negatives [7].

### 2.8. Model Interpretation and Feature Selection

Because the final ensemble consists of algorithmically different models, interpretation was performed per component using the most appropriate available method: SHAP TreeExplainer for CatBoost and LightGBM (both tree-based and SHAP-compatible) [8], native impurity-based feature_importances_ for RUSBoost (which is not supported by SHAP's TreeExplainer), and raw linear coefficients for SGDClassifier. The four resulting importance scores were min-max normalized and averaged to obtain a single combined importance score for each feature. Several feature-count thresholds were then evaluated. The retrained ensemble at each threshold was compared with the full

22-feature model using bootstrap resampling and McNemar's test before the final feature subset was selected.

## III. RESULTS AND DISCUSSION

### 3.1. Baseline and Individual Model Comparison

The DummyClassifier baseline achieved F1 = 0.0 and balanced accuracy = 0.5, as expected from a strategy that never predicts the positive class, establishing a clear lower bound against which all subsequent models were compared. Table 1 summarizes the validation performance of the five candidate models prior to hyperparameter tuning.

Table 1: Baseline Comparison of Five Candidate Models (Validation Set)

| Model | Validation F1 | Train F1 | Gap | Time (s) |
|---|---|---|---|---|
| CatBoost | **0.6464** | 0.9422 | 0.2958 | 17.950 |
| LightGBM | 0.6423 | 1.0000 | 0.3577 | 0.500 |
| Nearest Centroid | 0.6175 | 0.6599 | 0.0424 | 0.035 |
| RUSBoost | 0.6099 | 0.6707 | 0.0608 | 0.120 |
| SGDClassifier | 0.5051 | 0.5726 | 0.0675 | 0.100 |

CatBoost and LightGBM showed the largest train-validation gaps (0.2958 and 0.3577 respectively), indicating overfitting that motivated a regularization-focused hyperparameter search, while Nearest Centroid, RUSBoost, and SGDClassifier were substantially more stable by comparison.

### 3.2. Class Imbalance Handling

Applying model-specific imbalance handling improved validation F1 for four of the five models: CatBoost with scale_pos_weight improved from 0.6464 to 0.6803, LightGBM with balanced class weighting from 0.6423 to 0.6731, Nearest Centroid with empirical priors from 0.6175 to 0.6368, and SGDClassifier with balanced class weighting from 0.5051 to 0.5725. RUSBoost was the exception: adding sample weighting on top of its built-in random undersampling mechanism reduced F1 rather than improving it, and its unweighted default configuration was therefore retained.

### 3.3. Hyperparameter Tuning

Genetic Algorithm tuning produced measurable regularization effects consistent with the overfitting diagnosed in Section 3.1. CatBoost's tuned iteration count fell from a default of 1,000 to 67, with L2 regularization increasing from 3.0 to 8.44; LightGBM's tuned max_depth fell to 4 with reg_lambda increasing to 8.40. RUSBoost and SGDClassifier, which had shown smaller overfitting gaps, obtained the largest validation F1 improvements after tuning (+0.0497 and +0.0698 respectively), while CatBoost, LightGBM, and Nearest Centroid showed slightly reduced validation performance despite improved cross-validation scores. This finding illustrates that improvements in cross-validation performance do not necessarily translate into better validation performance when fold-specific patterns are exploited.

### 3.4. Ensemble Comparison

Among all 31 naive-averaging combinations, the best result (F1 = 0.6769) was obtained using all four probability-supporting models jointly, identical to the three-model combination excluding SGDClassifier, suggesting diminishing marginal contributions beyond the strongest models under equal weighting. Table 2 compares the three dedicated ensemble strategies against the best single model.

Table 2: Ensemble Strategy Comparison (Validation Set)

| Strategy | Validation F1 |
|---|---|
| CatBoost (best single model) | 0.6726 |
| Best of 31 naive combinations | 0.6769 |
| Stacking (Logistic Regression) | 0.6438 |
| Greedy Ensemble Selection | 0.6726 |
| **Weighted Voting (Dirichlet search)** | **0.6841** |

Weighted Voting achieved the highest validation F1 (0.6841), with an optimal weight vector of CatBoost = 0.0860, SGDClassifier = 0.2018, LightGBM = 0.0422, and RUSBoost = 0.6700. Notably, RUSBoost received the largest ensemble weight despite not being the strongest individual model, demonstrating that ensemble contribution is not necessarily proportional to standalone performance. Bootstrap resampling (1,000 iterations) showed Weighted Voting outperforming the best single model (CatBoost) on 81.6% of resampled sets (mean F1 0.6831 vs. 0.6716), supporting the statistical significance of the ensemble's advantage.

### 3.5. Threshold Optimization

An out-of-fold-optimized classification threshold of 0.5144 achieved F1 = 0.7034 on training data but only F1 = 0.6742 on the independent validation set, underperforming the conventional default threshold of 0.5 (F1 = 0.6841). This result was consistent across all data-split configurations explored during model development. It suggests that threshold optimization based on out-of-fold training predictions, although methodologically sound and free from data leakage, did not generalize reliably to this dataset. Therefore, the default threshold of 0.5 was retained.

### 3.6. Model Interpretation

RUSBoost's native feature importance was dominated by three of twenty-two features (temp_max = 0.6807, same-day BBQ_weather = 0.2862, pressure = 0.0331), consistent with its use of shallow decision stumps as weak learners. SHAP analysis of CatBoost and LightGBM showed a more distributed contribution pattern, with temp_max, temp_mean, and pressure consistently ranked highest; notably, pressure showed substantial SHAP contribution despite weak linear correlation with the target (Pearson $r$ = 0.063), indicating a nonlinear relationship not captured by linear correlation alone. SGDClassifier's linear coefficients ranked pressure above temp_max, illustrating that the linear

component of the ensemble captured different aspects of the feature-target relationship than the tree-based components.

### 3.7. Feature Selection and Statistical Verification

After combining all four interpretation sources into a single normalized importance score, a threshold retaining seven features (temp_max, pressure, temp_mean, same-day BBQ_weather, global_radiation, lagged global_radiation, and lagged pressure) achieved the highest validation F1 among all thresholds tested (0.6857), marginally exceeding the full 22-feature model (0.6841). Because the performance difference was marginal, bootstrap resampling (2,000 iterations) showed a win rate of only 55.5% for the 7-feature model, an inconclusive result that motivated a formal McNemar's test. The resulting contingency table showed agreement on 706 of 730 validation samples (574 jointly correct, 132 jointly incorrect), with the 7-feature and full models each uniquely correct on 13 and 11 samples respectively. The resulting statistic ($\chi^2 = 11.0,\ p = 0.8388$) provided no evidence of a statistically significant performance difference, and the 7-feature model was adopted as the final model on the basis of statistically equivalent performance combined with substantially improved parsimony and interpretability.

### 3.8. Final Model Evaluation on the Independent Test Set

The final model was retrained on the combined training and validation data (2,921 observations) using the seven selected features and evaluated once on the previously untouched 2008–2009 test set (731 observations). Table 3 reports the final performance metrics.

Table 3: Final model performance on the validation and independent test sets

| Metric | Validation | Test |
|---|---|---|
| F1-score | 0.6841 | 0.6776 |
| Precision | 0.5700 | 0.5700 |
| Recall | 0.8600 | 0.8400 |
| Accuracy | 0.8000 | 0.8100 |
| **ROC-AUC** | **0.8862** | **0.9076** |
| PR-AUC | 0.6855 | 0.6961 |

The gap between validation and test F1-score was only 0.0065, and ROC-AUC and PR-AUC on the test set were consistent with, or marginally better than, validation results. This small validation-test gap provides direct empirical evidence that the model's performance reflects genuine generalization rather than an artifact of repeated validation-set-based decision-making, supporting the feature-selection decision based on McNemar's test in Section 3.7. The precision-recall trade-off (moderate precision, high recall for the positive class) were consistent between validation and test sets, reflecting the intended effect of the class-weighting strategies applied during training. Two practical demonstration scenarios further illustrated model behavior. During a historically unambiguous period (27–31 December 2009), all five predictions matched observed outcomes with low predicted

probabilities (0.16–0.26), consistent with the strong seasonal signal identified during exploratory analysis. During a more ambiguous transitional period (1–10 July 2009), the model correctly predicted 7 of the 10 observations. Misclassifications occurred even for moderately high predicted probabilities (e.g., 0.65 on 10 July), illustrating the inherent uncertainty of next-day weather forecasting that was obscured by the original deterministic same-day label.

## IV. CONCLUSION

This study demonstrates that rigorous methodological safeguards substantially influence the interpretation of ensemble classification results on a real-world weather dataset. Discovering that the original same-day label was deterministic prevented the development of a spuriously perfect classifier and led to the reformulation of the problem as a genuine next-day forecasting task. A chronological three-way train-validation-test split, combined with bootstrap resampling and McNemar's test, revealed that (i) an out-of-fold-optimized classification threshold that appeared superior on training data failed to generalize to validation data, and (ii) a reduced 7-feature model was statistically indistinguishable from the full 22-feature model, justifying its adoption on parsimony grounds rather than validation performance alone. The final Weighted Voting, combining CatBoost, LightGBM, RUSBoost, and SGDClassifier with Dirichlet-optimized weights, achieved F1 = 0.6776 and ROC-AUC = 0.9076 on an independent test set never used during model development, with only a 0.0065 F1 gap from validation performance. This provides strong evidence that the proposed model generalizes well to unseen data.

Limitations of this study include its restriction to a single city (Basel) and a data collection period ending in early 2010. Therefore, the proposed model has not been evaluated against more recent climate conditions. Additionally, because ensemble weights and feature-selection decisions were both derived from the same validation set rather than an additional held-out validation layer, a residual risk of validation-specific overfitting cannot be entirely excluded. However, the small validation-test performance gap suggests that this risk is likely to be limited in practice. Future work could extend this methodology to other cities within the same dataset, incorporate probability calibration techniques such as isotonic regression, and evaluate alternative hyperparameter tuning strategies, including Bayesian optimization, to assess the sensitivity of the reported results to different tuning methods.

## REFERENCES

[1]. Caruana, R., Niculescu-Mizil, A., Crew, G., and Ksikes, A. 2004. "Ensemble Selection from Libraries of Models." Proceedings of the 21st International Conference on Machine Learning, 18.

[2]. Cawley, G.C. and Talbot, N.L.C. 2010. "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation." Journal of Machine Learning Research, 11, 2079-2107.

[3]. Chen, T. and Guestrin, C. 2016. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[4]. Dietterich, T.G. 2000. "Ensemble Methods in Machine Learning." Lecture Notes in Computer Science, 1857, 1-15.

[5]. Dorogush, A.V., Ershov, V., and Gulin, A. 2018. "CatBoost: Gradient Boosting with Categorical Features Support." arXiv preprint arXiv:1810.11363.

[6]. Ferri, C., Hernandez-Orallo, J., and Modroiu, R. 2009. "An Experimental Comparison of Performance Measures for Classification." Pattern Recognition Letters, 30(1), 27-38.

[7]. Kaufman, S., Rosset, S., Perlich, C., and Stitelman, O. 2012. "Leakage in Data Mining: Formulation, Detection, and Avoidance." ACM Transactions on Knowledge Discovery from Data, 6(4), 1-21.

[8]. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. 2017. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." Advances in Neural Information Processing Systems, 30, 3146-3154.

[9]. Klein Tank, A.M.G. and Coauthors. 2002. "Daily Dataset of 20th-Century Surface Air Temperature and Precipitation Series for the European Climate Assessment." International Journal of Climatology, 22, 1441-1453.

[10]. Kotz, S., Balakrishnan, N., and Johnson, N.L. 2000. "Continuous Multivariate Distributions, Volume 1: Models and Applications." 2nd ed. John Wiley & Sons, New York.

[11]. Lundberg, S.M. and Lee, S.-I. 2017. "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems, 30, 4765-4774.

[12]. McNemar, Q. 1947. "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages." Psychometrika, 12(2), 153-157.

[13]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. 2011. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 12, 2825-2830.

[14]. Saito, T. and Rehmsmeier, M. 2015. "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets." PLoS ONE, 10(3), e0118432.

[15]. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., and Napolitano, A. 2010. "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40(1), 185-197.