인문논총 83권 3호(2026. 8. 31.), pp. 297~325
[DOI] https://doi.org/10.17326/jhsnu.83.3.202608.297

Predicting the Size of Consonant Inventories through Machine Learning

Lee, Jinho*

ABSTRACT This study investigates how accurately machine learning can estimate consonant inventory sizes, or the number of consonant phonemes, across the world's languages. A typologically balanced dataset of 3,035 languages was divided into training (2,124, 70%), validation (455, 15%), and test (456, 15%) sets. Predictive models were built using 47 binary consonant variables alongside genealogical and geographical data. Among five algorithms compared, LightGBM performed best. On the test set, this model achieved a 13.7% error rate and an R^2 of 0.770. Furthermore, 24.8% of the languages had an error rate below 5.0%, and the predictions showed 61.2% agreement with the WALS-based five-level classification. Compared to baseline models, LightGBM attained high accuracy using only binary segmental features and limited extra-linguistic variables. Analysis revealed that prediction accuracy varied significantly across several factors but could not be explained by isolated consonants or simple co-occurrence patterns. The results suggest that the model's predictions were more likely driven by complex, multidimensional relationships among variables than by isolated predictors.

Keywords Consonant Inventory Size, Machine Learning, LightGBM, Predictor Variable, Feature Importance, Linguistic Typology

* Professor, Department of Korean Language and Literature, Seoul National University

1. Introduction

The size of consonant inventories varies considerably across languages. For example, Rotokas has only six consonant phonemes, whereas West !Xoon is reported to have more than 100 (Maddieson 1984; Gordon 2016).[1] Although phonemic analyses may vary depending on dialect or methodology, the extent of cross-linguistic variation in inventory size remains striking.

Factors associated with consonant inventory size are broadly categorized into extra-linguistic and linguistic correlates. Extra-linguistic factors such as region, population size, climate, language family, language contact, genetic variation, and community structure have frequently been cited as relevant, prompting substantial scholarly debate (Trudgill 2004; Hay and Bauer 2007; Atkinson 2011; Wichmann et al. 2011; Moran et al. 2012; Moran and Blasi 2014; Nikolaev 2019; Fenk-Oczlon and Pilz 2021). Linguistic factors include articulatory complexity, perceptual distinctiveness, syllable structure, and word length (Lindblom and Maddieson 1988; Nettle 1995; Maddieson 2007; Wichmann et al. 2011; Moran and Blasi 2014; Easterday 2019; Fenk-Oczlon and Pilz 2021).

However, the relationship between consonant inventory size and these variables remains insufficiently specified, particularly regarding extra-linguistic factors. This makes it difficult to predict inventory size from linguistic or extra-linguistic variables. Such prediction also requires quantifying diverse variables, using a large-scale language sample, and applying sophisticated predictive models. Until recently, these conditions had not been met.

This situation has now changed. Phoneme inventories for large-scale

---

1 The language names and family classifications are based on Hammarström et al. (2024).

samples are accessible through digital platforms, and advancements in machine learning offer powerful tools for phonological research. Given the availability of appropriate linguistic data, these developments make it possible to estimate phoneme inventory sizes using machine learning.

In this study, we compiled consonant inventories for over 3,000 languages and used machine learning to predict the number of consonant phonemes in each. Various models based on known correlates were then evaluated, and the best-performing algorithm was selected for analyzing prediction accuracy. The results show high predictive accuracy, indicating that consonant phoneme counts can be predicted from a limited number of variables. This approach supports, from a machine learning perspective, the traditional view that a phonological system is a complex structure rather than a mere sum of phonemes, and may also provide a basis for estimating overall phonemic inventory size in languages with fragmentary phonemic evidence or incomplete reconstructions.

2. Language Data

The importance of language data in linguistic typology cannot be overstated, as the success of a study depends largely on the quality and representativeness of the sample. For machine learning, datasets should ideally be large-scale and balanced across language families and geographic regions. Since constructing such balanced samples is labor-intensive, most typological studies on phoneme inventories rely on established datasets like UPSID or PHOIBLE.

While these datasets provide convenient access to large-scale phoneme

inventories, their limitations are well-documented. UPSID has been criticized for arbitrary allophone selection (Simpson 1999) and data inaccuracies (Vaux 2009; Moran 2012). PHOIBLE also faces scrutiny for its heterogeneity, stemming from diverse sources and potential biases toward specific language families or regions (Moran et al. 2021).

To address these issues, this study compiled a dataset based on Lee (2025), which provides bibliographic information and phonological analyses for 3,035 languages. From these materials, only consonant phonemes were extracted, with segment representations strictly aligned with IPA conventions. Language distribution by family and region was made proportional to Glottolog to minimize sampling bias. To ensure typological diversity, at least one language from every family with an available phonemic analysis was included. Although Lee (2025) is not without its limitations, the issues found in previous studies are considerably less pronounced.

The 3,035 languages in Lee (2025) represent 203 top-level families and 101 isolates. Most of the 247 families in Hammarström et al. (2024) are included, except for small families like Kwalean or Somahai, for which no phonemic analysis is available. Additionally, families with disputed classifications, such as Koreanic and Japonic, are treated as language isolates. The languages are distributed across seven macro-areas: Africa, Papunesia, Asia, South America, North America, Australia, and Europe. Although Glottolog groups Asia and Europe as Eurasia, this study treats them separately given their distinct phonological characteristics.

3. Design of the Prediction Method

3.1. Partitioning of Language Data

To ensure a rigorous machine learning framework and prevent data leakage, the dataset was partitioned into training (70%), validation (15%), and test (15%) sets. This process, including multi-variable stratification, was implemented in Python using the pandas and scikit-learn libraries.[2]

Table 1 reports the number of languages assigned to the training, validation, and test subsets under a 70 : 15 : 15 split. The subsets were mutually exclusive and collectively exhaustive, with each of the 3,035 languages assigned to exactly one subset. The stratified sampling procedure preserved the joint distribution of major language families and geographic areas. The maximum absolute deviation from the original distribution was 0.125 percentage points for the training set, 0.301 for the validation set, and 0.284 for the test set. Thus, the family-by-area proportions were consistently maintained across all subsets.

The three subsets served distinct purposes in model development and

[Table 1] Results of dataset partitioning

training	validation	test	total
2,124	455	456	3,035

2 All custom coding procedures in this study were developed with the assistance of Gemini (version 1.5 Pro) and ChatGPT (version 5.5 Pro). To ensure scientific rigor, the author independently verified all analyses, model outputs, and interpretations for accuracy and linguistic validity. Furthermore, to facilitate reproducibility, three core code scripts used for dataset partitioning and predictive modeling are provided in the Appendix.

evaluation. The training set was used to fit the models, while the validation set was used for hyperparameter tuning, comparing model configurations, and guiding model selection. The test set was held out from all training and validation procedures and reserved for the final evaluation, providing an independent assessment of how well the selected model generalizes to unseen languages.

3.2. Predictor Variables

In linguistic typology, selecting meaningful variables is crucial for valid comparisons and accurate predictions (Bickel 2007). This study excludes variables that are ambiguously defined or difficult to quantify reliably. Furthermore, because the analysis requires a large-scale dataset, factors attested in only a limited number of languages were deemed unsuitable as predictors.

This study incorporates language family, region, latitude, and longitude as extra-linguistic predictors. Language family and region are treated as categorical variables, while latitude and longitude are numerical. Linguistic predictors primarily consist of the presence or absence of individual consonants, each encoded as a binary variable. While the extra-linguistic variables are self-explanatory, the consonant-presence variables require further clarification.

In the 3,035-language sample, over 1,000 consonant phonemes were identified, and those with a significant correlation to inventory size were selected as key features. This correlation was measured using the Pearson Correlation Coefficient (PCC) under two conditions. First, PCC values were calculated solely from the training subset, effectively precluding data

leakage by keeping validation and test information isolated. Second, target consonants were limited to those occurring in at least 1% of the training languages (i.e., present in 21 or more languages). This criterion helps ensure statistical stability by maintaining a sufficient sample size.

However, the strict application of this second condition would exclude all click consonants. Given their strong association with exceptionally large inventories despite their low frequency, their wholesale exclusion is unjustifiable. Yet, including individual click phonemes is infeasible due to data sparsity. To address this, rather than distinguishing each phoneme, we aggregated them into a single binary feature to indicate their presence or absence and evaluated its PCC.

In the training dataset, 157 consonants met the threshold of being attested in at least 21 languages. Including the single binary click-presence feature, a total of 158 consonantal features were identified. PCC values were calculated between the presence of each feature and consonant inventory size. Table 2 presents the highest-ranked features, categorized into the top 10% (16), 20% (32), and 30% (47). The optimal percentage threshold for these variables was determined in 3.3 through a sensitivity analysis.[3]

All PCC values in Table 2 are presented as absolute values. Of the 158 features, 152 show a positive correlation, while 6 exhibit a negative one. For the latter, PCC values are ≥ -0.112; thus, no consonant in Table 2 exhibits a meaningful negative correlation with consonant inventory size. All PCC values in Table 2 are statistically significant ($p < .0001$).

Some may argue that predicting inventory size from the presence

3 Since this study does not aim to maximize prediction accuracy, the sensitivity analysis regarding the consonant variable ratio was not conducted in extensive detail.

[Table 2] Consonants as predictor variables for machine learning

0~10%		10~20%		20~30%	
Consonants	PCC	Consonants	PCC	Consonants	PCC
kʰ	0.406	tʰ	0.277	v	0.248
pʰ	0.376	qʼ	0.277	ʑ	0.247
ʤ	0.369	qʼʷ	0.276	tʼ	0.243
click	0.361	ʈʰ	0.274	qʰ	0.243
z	0.355	ʧʼ	0.274	ʦʰ	0.242
tsʰ	0.322	χʷ	0.273	f	0.240
χ	0.320	g	0.272	ʁ	0.238
ʒ	0.320	ʡʰ	0.263	ɕ	0.235
ʦ	0.310	kʼʷ	0.263	ʈʂ	0.234
ɬ	0.307	ɖʐ	0.261	pʲ	0.227
tsʼ	0.306	qʷ	0.258	ɣ	0.227
ʧʰ	0.301	x	0.253	ʧ	0.225
ʃ	0.298	ʤ	0.251	tɕ	0.225
kʼ	0.293	tɕʰ	0.249	kʷ	0.224
gʷ	0.278	pʼ	0.249	b	0.223
q	0.278	tʷ	0.249		

of specific consonants is circular or trivial. This assumes that simple summation equals prediction, which is unconvincing. However, as Table 5 in 3.3 shows, the simple sum of these variables has limited accuracy and performs substantially worse than non-linear machine learning models. This suggests that the predictive gain may arise not simply from counting consonants, but from modeling more complex relationships among genealogical, areal, geographic, and segmental features. Thus, the present approach is neither circular nor trivial, as its predictive performance cannot be reduced to simple summation.

3.3. Machine Learning Procedures and Predictive Model Selection

To identify the most effective predictive approach, we employed five machine learning models: Random Forest, Poisson Regression, Ridge Regression, LightGBM, and K-Nearest Neighbors. These were selected to cover diverse predictive strategies, as their unique algorithmic characteristics allow us to assess how different frameworks handle the data structure. Specifically, we include linear methods (Poisson and Ridge Regression) to capture additive effects, a distance-based method (K-Nearest Neighbors) to identify local similarities, and ensemble techniques (Random Forest and

[Table 3] Comparative performance by model and consonant variable ratio

Models		MAE (↓)	RMSE (↓)	R^2 (↑)
LightGBM	30%	**3.053**	4.753	0.760
	20%	3.215	5.116	0.722
	10%	3.631	5.516	0.676
Random Forest	30%	3.084	4.861	0.749
	20%	3.218	5.070	0.727
	10%	3.442	5.261	0.706
Ridge Regression	30%	3.231	**4.706**	**0.765**
	20%	3.442	5.044	0.729
	10%	3.738	5.573	0.670
K-Nearest Neighbors	30%	3.484	5.534	0.674
	20%	3.462	5.556	0.672
	10%	3.521	5.469	0.682
Poisson Regression	30%	3.521	5.350	0.696
	20%	3.736	5.724	0.652
	10%	4.029	6.143	0.599

LightGBM) to accommodate non-linear complexities. By evaluating these heterogeneous models, this study systematically explores the predictive logic best suited to our linguistic dataset, avoiding reliance on a potentially restrictive analytical approach. We also performed sensitivity analyses by varying the consonant variable ratios (10%, 20%, and 30%) for each model to evaluate their stability.

Validation set evaluation revealed that predictive performance generally improved with the consonant variable ratio, except for K-Nearest Neighbors. Consequently, the top 30% was selected as the final ratio. Under this condition, models were compared using MAE, RMSE, and R^2. LightGBM achieved the lowest MAE, while Ridge Regression performed best in RMSE and R^2. Since this study prioritizes minimizing the absolute error in the number of consonant phonemes, LightGBM was selected as the final model.[4] Nevertheless, Ridge Regression's superior performance in RMSE and R^2 warrants its inclusion as a supplementary model for the test set.

In Table 4, predictive performance on the test set mirrors the validation results. The final model, LightGBM, achieved an MAE of 3.033, an RMSE

[Table 4] Predictive performance of the primary and supplementary models on the test set

Models	MAE (↓)	RMSE (↓)	R^2 (↑)
☞ LightGBM	**3.033**	4.522	0.770
Ridge Regression	3.215	**4.457**	**0.776**

4 Although the MAE difference between LightGBM and Ridge Regression was modest, a paired absolute-error comparison across validation languages showed a statistically significant advantage for LightGBM (Wilcoxon signed-rank test, two-sided p = .0101). A similar pattern emerged in the test-set MAE values shown in Table 4 (two-sided p = .0137).

of 4.522, and an R^2 of 0.770. These results remain consistent with Table 3: LightGBM outperforms the supplementary Ridge Regression model in MAE, while showing slightly lower performance in RMSE and R^2. LightGBM's predictive superiority is further substantiated by its comparison with the baseline models.

To establish a baseline, we implemented four benchmark models. Each was parameterized using the training set and evaluated on the test set. As summarized in Table 5, the first three models use categorical averages as reference values: (a) the global mean of consonant inventories, (b) the mean per macro-area, and (c) the mean per language family. The fourth baseline (d) is derived from the sum of consonant variables. Given that the raw sum deviates from actual inventory sizes, we applied linear regression to calibrate predictions. The adjusted model is defined as 'Y = 12.98+1.53 × S', where 'S' denotes the sum of consonant variables.[5]

As presented in Table 5, baseline models utilizing categorical averages yield unsatisfactory predictive results. The calibrated sum-based model (d) demonstrates the highest performance; however, its predictive power remains substantially lower than that of the LightGBM model (Table 4).

[Table 5] Predictive performance of four baseline models

	Baseline	MAE (↓)	RMSE (↓)	R^2 (↑)
(a)	Overall Mean	7.12	9.440	-0.003
(b)	Macro-area Mean	5.65	7.826	0.311
(c)	Family Mean	5.19	7.165	0.422
(d)	Calibrated Segmental-Sum	4.14	5.863	0.613

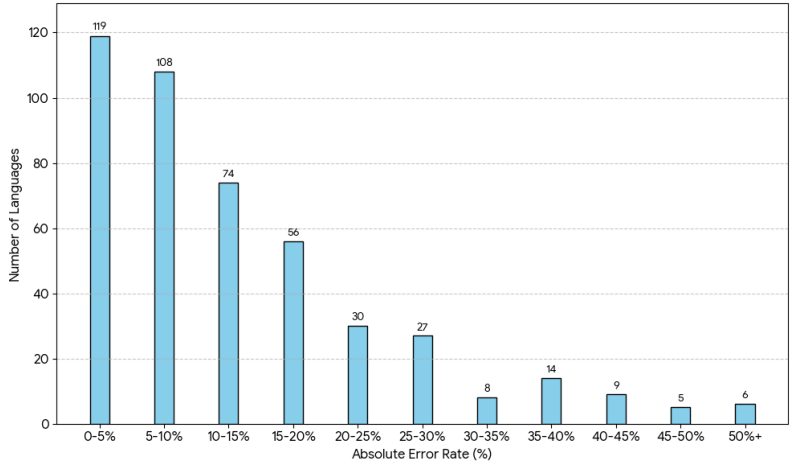
5 The values 12.98 and 1.53 are, respectively, the intercept and slope estimated by fitting the linear regression model to the training data.

LightGBM outperforms all hierarchical baselines, including overall, macro-area, and family means, as well as the sum-based linear model. This indicates that non-linear learning effectively captures complex phonological patterns that traditional mean-based or simple linear approaches fail to identify.

4. Prediction Results

Since consonant inventory sizes must be integers, the decimal outputs of the LightGBM model were rounded to the nearest whole number. Prediction accuracy was subsequently calculated based on the error rate between actual and predicted values. Figure 1 illustrates the distribution of languages according to this error rate.

As shown in Figure 1, the largest number of languages exhibits an error rate of 5% or less. Furthermore, nearly half of the 456 languages in the



[Figure 1] Distribution of languages by prediction error rate

test dataset have an error rate of 10% or less. Conversely, there are only 42 languages with an error rate exceeding 30%, which accounts for less than 10% of the test set. The average error rate across all languages is 13.7%.

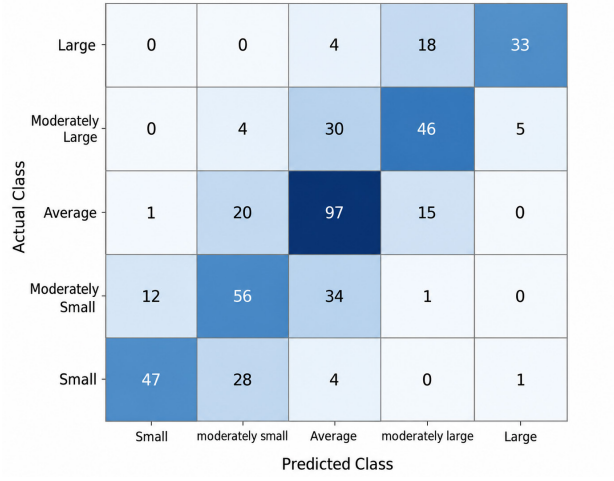
The model predicted the exact consonant count for 65 languages, representing 14.3% of the test set. These range from small inventories, such as Pukapuka (10), to large ones, such as Wutunhua (38) and Kako (33). The highest error rate occurs in Mbabaram, an Australian language; the model predicted 35 consonants for this 14-consonant language, yielding an error rate of 150.0%. This outlier does not appear to reflect a broader effect of geographic region or language family. Excluding this case, the average error rates for Australian languages and the Pama-Nyungan family are 10.5% and 7.4%, respectively, both below the overall average.

The error rate does not appear to be related to the number of consonants. Contrary to the expectation that a larger inventory would lead to a higher error rate, there is virtually no correlation between the number of consonant phonemes and the error rate. The Pearson correlation coefficient was −0.043 ($p = .358$), which exceeds the significance level of .05. In other words, there is no statistically significant linear correlation between the number of consonants and the prediction error rate.

To facilitate subsequent discussion, prediction accuracy is categorized into three levels (High, Moderate, and Low) based on the error rate. Using quartiles, these categories are defined as summarized in Table 6.

[Table 6] Prediction accuracy grades by error rate

Accuracy grades	High	Moderate	Low
Error rate interval	0~4.9%	5.0~18.8%	18.9%~
Mean error rate	1.7%	10.7%	32.0%
Languages	113	230	113



[Figure 2] Confusion matrix of WALS-based consonant size grades

Next, prediction accuracy is examined based on the WALS five-level classification: small (6–14), moderately small (15–18), average (19–25), moderately large (26–33), and large (34 and above). For the LightGBM model, the predicted grades exactly matched actual grades in 279 out of 456 languages, yielding an accuracy of 61.2%.

Accuracy by grade was relatively high for the average category, while the other categories showed similar levels of accuracy.

(1) Prediction accuracy by WALS grade

a. Small: 47 out of 80 languages (58.8%)

b. Moderately small: 56 out of 103 (54.4%)

c. Average: 97 out of 133 (72.9%)

d. Moderately large: 46 out of 85 (54.1%)

e. Large: 33 out of 55 (60.0%)

While a 61.2% classification accuracy may appear modest compared to the 13.7% error rate and 0.770 R^2, applying Quadratic Weighted Kappa (QWK) provides a broader perspective. Unlike simple accuracy, QWK penalizes larger deviations more heavily. The QWK score of 0.812 falls into the "Almost perfect" agreement category as defined by Landis and Koch (1977).[6]

5. Analysis of Prediction Results

5.1. Contribution of Variable Groups to Prediction

We predicted consonant inventory size using extra-linguistic variables (language family, macro-area, and geographical coordinates) alongside 47 intra-linguistic consonant-presence features. To evaluate the contribution of each variable group, a leave-one-group-out ablation analysis was performed using the LightGBM model. In each iteration, one group was excluded while the model was retrained on the same training data using identical hyper-parameters. Predictive performance was then evaluated on the independent test set using MAE, RMSE, and R^2 and compared against the full model.

Table 7 reveals a clear trend in predictive performance. Omitting any single extra-linguistic variable group results in negligible degradation compared to the full model. For instance, excluding language family causes

6 The Landis and Koch (1977) benchmarks were developed for categorical data. Given that this study discretizes consonant inventory sizes to align with the WALS five-level classification, these benchmarks serve as a heuristic indicator of strong ordinal agreement rather than an absolute categorical metric.

[Table 7] Predictive performance after removing each variable group

Removed group	MAE (↓)	RMSE (↓)	R^2 (↑)
None	3.033	4.522	0.770
Language family	3.055	4.541	0.768
Macro-area	3.011	4.515	0.771
latitude/longitude	3.193	4.631	0.759
Consonants	4.357	5.992	0.596

only minor deterioration in MAE (+0.022), RMSE (+0.019), and R^2 (−0.002), while omitting the macro-area variable yields a marginal improvement in accuracy. Conversely, excluding the 47 intra-linguistic consonant-presence features leads to a substantial decline across all metrics. These findings demonstrate that intra-linguistic features contribute significantly more to predicting consonant inventory size than the examined extra-linguistic variables.

5.2. Analysis by Language Family

The 456 test languages represent 104 top-level language families. To ensure statistical stability, the analysis was restricted to major language families, defined as those represented by at least 10 languages in the test set. As shown in Table 8, eight families meet this criterion.

Table 8 displays the distribution of accuracy grades by language family. Predictive accuracy varies substantially: the proportion of High-grade predictions ranges from 42.9% for Afro-Asiatic to only 8.3% for Nuclear Trans New Guinea. Conversely, Low-grade predictions account for 7.7% of Austroasiatic languages, but 50.0% of Nuclear Trans New Guinea languages. This pattern is mirrored in the mean error rates: Austroasiatic

[Table 8] Proportion of accuracy grades by language family

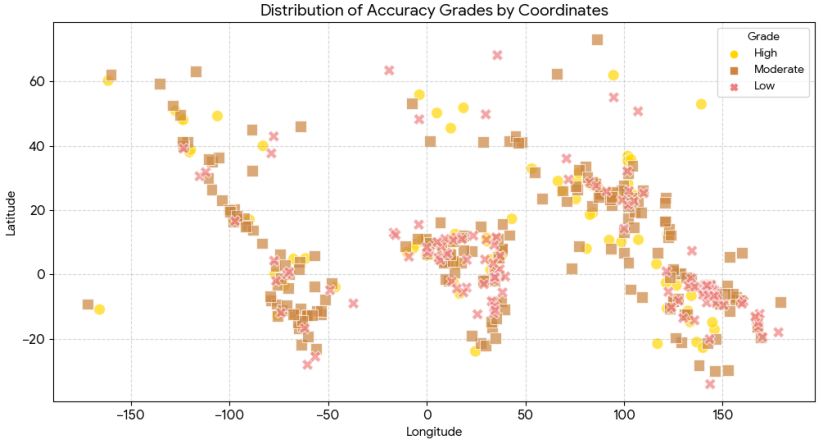
Families	Languages	High	Moderate	Low	error rate
Atlantic-Congo	78	21.8% (17/78)	42.3% (33/78)	35.9% (28/78)	16.1%
Austronesian	65	18.5% (12/65)	58.5% (38/65)	23.1% (15/65)	12.9%
Sino-Tibetan	39	35.9% (14/39)	46.2% (18/39)	17.9% (7/39)	11.3%
Indo-European	28	39.3% (11/28)	42.9% (12/28)	17.9% (5/28)	10.3%
Afro-Asiatic	21	42.9% (9/21)	33.3% (7/21)	23.8% (5/21)	10.9%
Pama-Nyungan	14	35.7% (5/14)	50.0% (7/14)	14.3% (2/14)	17.6%
Austroasiatic	13	30.8% (4/13)	61.5% (8/13)	7.7% (1/13)	9.9%
Nuclear Trans New Guinea	12	8.3% (1/12)	41.7% (5/12)	50.0% (6/12)	21.3%

exhibits the lowest at 9.9%, compared with 21.3% for Nuclear Trans New Guinea. Collectively, these results indicate that Nuclear Trans New Guinea shows the lowest predictive accuracy among the major families, whereas Austroasiatic yields comparatively favorable results.

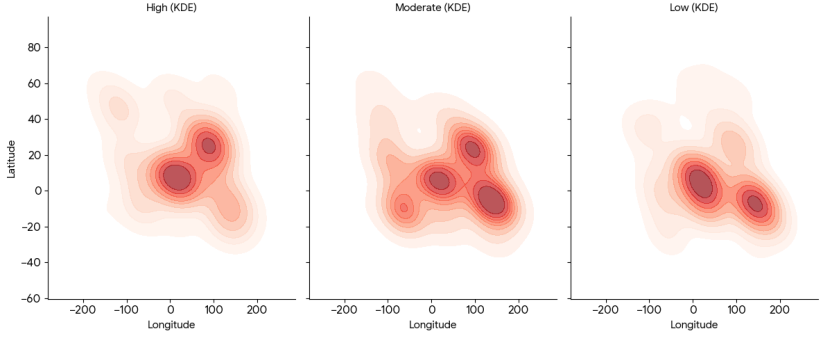
5.3. Analysis by Geographic Area

This section examines prediction accuracy relative to language coordinates. Figures 3 and 4 visualize the spatial distribution and geographic density of languages across the three accuracy levels, illustrating geographical variations in model performance.

Figure 3 shows that the three grades are broadly intermingled globally,



[Figure 3] Geographic distribution of languages by prediction accuracy grade



[Figure 4] Geographic density of languages by prediction accuracy grade

with no clear geographic separation or concentration unique to any accuracy level. Languages of different grades coexist within major linguistic regions, including West Africa, island Southeast Asia, and New Guinea. Furthermore, the kernel density estimates in Figure 4 demonstrate that all three grades share highly similar geographic concentration patterns. Although minor local differences are observable, principal density peaks consistently occur in the same regions.[7] These regions coincide with areas

of high linguistic diversity and dense sampling, suggesting that the observed spatial patterns primarily reflect the distribution of sampled languages rather than systematic geographic effects on prediction accuracy.

5.4. Analysis by Consonant Features

As the presence or absence of specific consonants was the primary predictor, the relationship between prediction outcomes and consonants can be examined from multiple perspectives.

(2) Correlation between prediction results and consonant variables

a. Feature importance of consonant variables

b. Relationship between consonant variables and prediction accuracy

c. Skewness of consonant variables toward specific accuracy grades

(2a) examines the contribution of each consonant variable listed in Table 2 to the prediction process. Table 9 presents the gain-based feature importance (Gain) from the final LightGBM model alongside its normalized values (Gain %). This percentage represents each feature's relative contribution to the total Gain across all 47 consonant variables.

Table 9 presents the feature importance of the consonant variables. We focus on the top ten consonants at the 'elbow' of the Gain values, where contributions transition from steep to gradual decline. Collectively, these ten

---

7 The three local density peaks for the Moderate grade, compared to two for the High and Low grades, should be interpreted cautiously. This discrepancy likely arises from the larger sample size of the Moderate group (n = 230) compared to the others (n = 113 each), which may allow the KDE to identify finer clusters.

[Table 9] Feature importance of consonant variables by gain method

Rank	Consonants	Gain	Gain %	Rank	Consonants	Gain	Gain %
1	k^h	205484	17.9	6	t^w	53663	4.7
2	click	109316	9.5	7	ʤ	49404	4.3
3	z	94662	8.3	8	p^j	47888	4.2
4	q'^w	59816	5.2	9	ʒ	39530	3.4
5	k^w	59079	5.2	10	q^w	30882	2.7

consonants account for 65.4% of total Gain, demonstrating their dominant contribution. Among them, /k^h/ exhibits the highest contribution, followed by click consonants and /z/.

Two additional Pearson correlation analyses examined the relationship between consonant feature importance (Gain %) and other variables: the PCC values from Table 2 and consonant frequency in the training set. Gain (%) significantly correlated with PCC values ($r = 0.559$, $p < .001$), while its correlation with frequency was negligible ($r=0.054$, $p = .720$). These results suggest that consonants more strongly associated with inventory size carried greater feature importance. Conversely, feature importance showed no meaningful association with the simple frequency of a consonant's appearance in the training sample.

(2b) concerns the correlation between the consonant variables and prediction accuracy. This can be further subdivided into the following three aspects.

(3) Relationship between Consonant Variables and Prediction Accuracy

a. Correlation between the presence or absence of consonant variables and prediction accuracy

b. Correlation between the number of consonant variables and prediction

accuracy

c. Correlation between the co-occurrence patterns of consonant variables and prediction accuracy

(3a) examines the relationship between the 47 consonant variables and prediction accuracy. The highest Pearson correlation coefficient (PCC) between consonant presence/absence and error rate was 0.304 (for /t^{w}/, $p < .001$), with all others remaining within ± 0.128. Furthermore, except for a few consonants including /t^{w}/, p-values largely exceed 0.05, indicating that most correlations were not statistically significant. These results suggest that prediction errors were not strongly or systematically associated with the presence or absence of individual consonants.

(3b) addresses the correlation between the total count of consonant variables and error rate. Since languages contain varying numbers of these variables, examining their impact on accuracy is necessary. Given the low predictive accuracy of the Calibrated Segmental-Sum in Table 5, a strong correlation was not anticipated. The analysis yielded a PCC of −0.072 ($p = .124$), indicating no meaningful correlation. This lack of a meaningful correlation persists even when limiting the analysis to the ten most important consonants from Table 9 (PCC = 0.043, $p = .356$).

(3c) examines whether co-occurrence patterns of consonant variables relate to prediction accuracy. For each of the 1,081 pairs formed from the 47 variables, the co-occurrence rate was calculated as the proportion of languages containing both consonants. Pairs never observed in the data ($n = 203$) were excluded. Subsequently, the mean prediction error rate was computed for languages containing each pair. Finally, Pearson's correlation coefficient tested whether co-occurrence rates were associated with predic-

[Table 10] Pearson correlations between consonant-pair co-occurrence rate and mean error rate under different inclusion thresholds

Inclusion thresholds	Consonant pairs	Pearson's r	p-value
≥ 1 language	878	-0.051	.134
≥ 5 languages	578	0.127	.002
≥ 10 languages	335	0.211	< .001

tion error rates, with results presented in Table 10.

Due to many consonant pairs appearing in a limited number of languages, Pearson's correlations were calculated under three inclusion thresholds: pairs attested in at least 1, 5, and 10 languages. Including all attested pairs revealed no significant correlation between co-occurrence rate and mean prediction error. However, excluding rare instances yielded weak but significant positive correlations for pairs attested in at least 5 (r = 0.127, p = .002) and 10 languages (r = 0.211, p < .001). These results suggest that the relationship between consonant-pair co-occurrence and prediction error is sensitive to rare-case treatment and remains relatively weak even under stricter thresholds.

Finally, (2c) examines whether individual consonants exhibit uneven distributions across the three accuracy grades. For each of the 47 variables, a 2×3 contingency table crossed consonant presence with accuracy grade. Chi-square tests of independence assessed the relationship between consonant occurrence and accuracy grade. Since 47 tests were performed, p-values were adjusted using the Benjamini–Hochberg false discovery rate procedure. For significant associations, adjusted standardized residuals identified overrepresented accuracy grades; residuals exceeding +1.96 indicated significant overrepresentation. Effect size was measured using Cramer's V.

Although three consonants (/b/, /χ^w/, /f/) showed nominally significant associations with accuracy grades prior to correction, none remained significant after Benjamini–Hochberg FDR correction. This indicates that no individual consonant exhibited a statistically robust skew toward any specific accuracy grade. Consequently, the distribution of consonant-presence variables across accuracy grades provides no evidence that specific consonants systematically drive prediction accuracy differences.

The variables most crucial to the machine learning, as revealed in Table 7, were the consonant-presence features selected for their correlations with inventory size. However, this section found no consistent associations between prediction accuracy and individual consonants or pairwise co-occurrence patterns. These findings suggest that the predictive contribution of consonant-presence features is not reducible to such local segmental effects. Rather, the results are consistent with the possibility that the model derives its predictions from broader, multidimensional configurations among the consonant variables. Nevertheless, the partially opaque nature of the model remains a fundamental limitation in fully elucidating the exact non-linear interactions exploited.

6. Conclusion

This study constructed a dataset from large-scale consonant inventories to investigate the accuracy of machine learning models in estimating phoneme counts across languages. The primary aim was not merely to maximize performance, but to evaluate the feasibility of such approximations under appropriate procedures. Despite limited model capabilities, the model

achieved higher accuracy than baselines. The evidence that the algorithm avoids reliance on isolated variables aligns with the fundamental linguistic premise that phonological systems are not random, but instead exhibit systematic patterns.

A highly predictive machine learning model could open the door to a range of applications. One such possibility is its use in evaluating the validity of reconstructed consonant systems. If model predictions diverge substantially from reconstructed values, this may require renewed scrutiny. Furthermore, a model like the one presented here, which can estimate the number of consonant phonemes from a limited set of information, might provide supplementary assistance to linguistic research, which often must rely on incomplete data.

References

Atkinson, Q. D. 2011. "Phonemic diversity supports a serial founder effect model of language expansion from Africa." *Science* 332(6027): 346–9.

Bickel, B. 2007. "Typology in the 21st century: Major theoretical developments." *Linguistic Typology* 11(1): 239–51.

Easterday, S. 2019. *Highly Complex Syllable Structure: A Typological and Diachronic Study*. Berlin: Language Science Press.

Fenk-Oczlon, G. and J. Pilz. 2021. "Linguistic complexity: Relationships between phoneme inventory size, syllable complexity, word and clause length, and population size." *Frontiers in Communication* 6: 1–7.

Gordon, M. K. 2016. *Phonological Typology*. Oxford: Oxford University Press.

Hammarström, H., R. Forkel, M. Haspelmath and S. Bank. 2024. *Glottolog* 5.1. Leipzig: Max Planck Institute for Evolutionary Anthropology. http://glottolog.org (Accessed: 2025. 4. 21.).

Hay, J. and L. Bauer. 2007. "Phoneme inventory size and population size." *Language* 83(2): 388–400.

Landis, J. R. and G. G. Koch. 1977. "The measurement of observer agreement for

categorical data." *Biometrics* 33(1): 159–74.

Lee, J. 2025. *A Bibliography of Phonological Sources for Typological Research*. Seoul: Jipmoondang.

Lindblom, B. and I. Maddieson. 1988. "Phonetic universals in consonant systems." in Hyman, L. M. and Charles, N. L. (eds) *Language, Speech and Mind: Studies in Honour of Victoria A. Fromkin*, 62–78. London: Routledge.

Maddieson, I. 1984. *Patterns of Sounds*. Cambridge: Cambridge University Press.

Maddieson, I. 2007. "Interrelationships between Measures of Phonological Complexity–Statistical Analysis of the Relationship between Syllable Structures, Segment Inventories, and Tone Contrasts–." in Solé, M., Beddor, P. S. and Ohala, M. (eds) *Experimental Approaches to Phonology*, 93–103. Oxford: Oxford University Press.

Moran, S. 2012. "Phonetics Information Base and Lexicon (PHOIBLE)." PhD diss., University of Washington.

Moran, S., D. McCloy and R. Wright. 2012. "Revisiting population size vs. phoneme inventory size." *Language* 88(4): 877–93.

Moran, S. and D. Blasi. 2014. "Cross-linguistic comparison of complexity measures in phonological systems." in Newmeyer, F. J. and Preston, L. B. (eds) *Measuring Grammatical Complexity*, 217–40. Oxford: Oxford University Press.

Moran, S., N. A. Lester, and E. Grossman. 2021. "Inferring recent evolutionary changes in speech sounds." *Philosophical Transactions of the Royal Society B: Biological Sciences* 376(1824): 20200198.

Nettle, D. 1995. "Segmental inventory size, word length, and communicative efficiency." *Linguistics* 33(2): 359–67.

Nikolaev, D. 2019. "Areal dependency of consonant inventories." *Language Dynamics and Change* 9(1): 104–26.

Simpson, A. P. 1999. "Fundamental problems in comparative phonetics and phonology: Does UPSID help to solve them?" in *Proceedings of the 14th International Congress of Phonetic Sciences*: 349–52.

Trudgill, P. 2004. "Linguistic and social typology: The Austronesian migrations and phoneme inventories." *Linguistic Typology* 8(3): 305–20.

Vaux, B. 2009. "The role of features in a symbolic theory of phonology." in Raimy, E. and Cairns, C. E. (eds) *Contemporary Views on Architecture and Representations in Phonology*, 74–97. Cambridge: MIT Press.

Wichmann, S., T. Rama, and E. W. Holman. 2011. "Phonological diversity, word length, and population sizes across languages: The ASJP evidence." *Linguistic Typology* 15(2): 177–97.

원고 접수일: 2026년 7월 11일, 심사완료일: 2026년 7월 24일, 게재 확정일: 2026년 8월 10일

초록

기계 학습을 통한 자음 체계의 크기 예측

이진호*

이 연구에서는 기계 학습을 활용하여 전 세계 언어의 자음 목록 크기를 얼마나 정확하게 예측할 수 있는지 살펴보았다. 유형론적으로 균형 잡힌 3,035개 언어의 자음 음소 목록을 훈련용(2,124개, 70%), 검증용(455개, 15%), 예측용(456개, 15%)의 세 부류로 나누어 분석하였다. 47개의 자음 유무 및 계통적/지리적 정보를 결합하여 예측 모델을 구축했으며, 훈련용과 검증용 자료를 대상으로 다섯 가지 기계 학습 모델을 평가하여 가장 우수한 성능을 보인 LightGBM을 최종 모델로 채택했다. 이 모델을 예측용 자료에 적용한 결과, 오차율 13.7%, 결정 계수(R^2) 0.770으로서 몇몇 기준 모델(baseline)과 비교해 높은 정확도를 보였다. 전체 언어의 24.8%가 오차율 5.0% 미만인 '우수'(high-grade) 범주에 해당했으며, WALS 5단계 분류와의 일치도는 61.2%였다. 이는 몇몇 언어 외적 변수 및 일부 자음의 유무만으로도 자음 음소의 개수를 상당히 정확한 수준으로 예측할 수 있음을 말해 준다. 예측력의 수준은 변수에 따라 상당한 차이를 드러냈으며, 개별 자음들의 존재 여부나 단순한 공기 관계만으로 설명되지는 않았다. 이로 미루어 볼 때 기계 학습을 통한 자음 개수의 예측은 변수들 사이의 복잡 다면한 관계를 더 중시했을 가능성이 높다.

주제어 자음 목록 크기, 기계 학습, LightGBM, 예측 변수, 기여도, 언어 유형론

* 서울대학교 국어국문학과 교수

〈Appendix〉 Core Code Collection

(1) Data partitioning code

```
import pandas as pd
from sklearn.model_selection import train_test_split
FILE_NAME = "source_data.xlsx"
df = pd.read_excel(FILE_NAME, header=None)
header = df.iloc[:1]
data = df.iloc[2:].copy()
columns = df.iloc[0].astype(str).str.strip()
family = columns[columns == "대어족"].index[0]
area = columns[columns == "area"].index[0]
data[family] = data[family].astype(str).str.strip()
data[area] = data[area].astype(str).str.strip()
data["strata"] = data[family] + "_" + data[area]
counts = data["strata"].value_counts()
data.loc[data["strata"].isin(counts[counts < 3].index), "strata"] = "Other"
train, temp = train_test_split(data, test_size=0.30, random_state=42, stratify=data["strata"])
temp = temp.copy()
counts = temp["strata"].value_counts()
temp.loc[temp["strata"].isin(counts[counts < 2].index), "strata"] = "Other"
if temp["strata"].value_counts().min() < 2:
    temp["strata"] = temp["strata"].where(temp["strata"].map(temp["strata"].value_counts()) >= 2, temp["strata"].mode()[0])
val, test = train_test_split(temp, test_size=0.50, random_state=42, stratify=temp["strata"])
for name, part in {"train": train, "val": val, "test": test}.items():
    part = part.drop(columns="strata")
    pd.concat([header, part], ignore_index=True).to_excel(f"{name}_data.xlsx", index=False, header=False)
```

(2) Training and validation code

```
from pathlib import Path
import json, zipfile, joblib
import numpy as np
import pandas as pd
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.linear_model import Ridge, PoissonRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from lightgbm import LGBMRegressor
BASE_DIR = Path("/content")
FILES = {
    "10%": ("train_data_70(변수10%).xlsx", "val_data_15(변수10%).xlsx", 16),
    "20%": ("train_data_70(변수20%).xlsx", "val_data_15(변수20%).xlsx", 32),
    "30%": ("train_data_70(변수30%).xlsx", "val_data_15(변수30%).xlsx", 47)
}
TARGET, LANG, LAT, LON, FAMILY, AREA = "자음", "언어", "위도", "경도", "대어족", "area"
CAT_COLS, NUM_COLS = [FAMILY, AREA], [LAT, LON]
RANDOM_STATE = 42
def onehot():
    try:
        return OneHotEncoder(handle_unknown="ignore", sparse_output=False)
    except TypeError:
        return OneHotEncoder(handle_unknown="ignore", sparse=False)
def load_data(path):
    df = pd.read_excel(path, engine="openpyxl")
    df.columns = [str(c).strip() for c in df.columns]
    df = df.iloc[2:].copy()
    df[TARGET] = pd.to_numeric(df[TARGET], errors="coerce")
    df = df[df[TARGET].notna()].copy()
    for col in NUM_COLS:
        df[col] = pd.to_numeric(df[col].astype(str).str.replace(",", ".", regex=False), errors="coerce")
    for col in CAT_COLS:
        df[col] = df[col].astype("string").fillna("Unknown")
```

```
        return df.reset_index(drop=True)
    def integerize(x):
        return np.clip(np.rint(x), 0, None).astype
(int)
    def scores(y, pred):
        return {
            "MAE": mean_absolute_error(y, pred),
            "RMSE": np.sqrt(mean_squared_error
(y, pred)),
            "R2": r2_score(y, pred)
        }
    def models():
        return {
            "Random Forest":
RandomForestRegressor(n_estimators=500,
random_state=RANDOM_STATE, n_jobs=-1),
            "Poisson Regression": PoissonRegressor
(alpha=1.0, max_iter=3000),
            "Ridge Regression": Ridge(alpha=1.0),
            "LightGBM": LGBMRegressor(n_
estimators=500, learning_rate=0.05, random_
state=RANDOM_STATE, n_jobs=-1, verbosity
=-1),
            "KNN Regression":
KNeighborsRegressor(n_neighbors=5, weights=
"distance")
        }
    model_dir = BASE_DIR / "trained_models_
validation"
    model_dir.mkdir(exist_ok=True)
    results, predictions, saved = [], [], {}
    for ratio, (train_file, val_file, n_cons) in
FILES.items():
        train = load_data(BASE_DIR / train_file)
        val = load_data(BASE_DIR / val_file)
        cons_cols = [c for c in train.columns if c
not in [TARGET, LANG, LAT, LON, FAMILY,
AREA]]
        if len(cons_cols) != n_cons:
            raise ValueError(f"{ratio}: expected {n_
cons} consonant variables, found {len(cons_
cols)}")
        features = CAT_COLS + NUM_COLS +
cons_cols
        X_train, y_train = train[features], train
[TARGET].astype(float)
        X_val, y_val = val[features], val
[TARGET].astype(float)
        preprocessor = ColumnTransformer([
            ("cat", onehot(), CAT_COLS),
            ("geo", StandardScaler(), NUM_COLS),
            ("cons", "passthrough", cons_cols)
        ])
        for name, model in models().items():
            pipe = Pipeline([("preprocess",
preprocessor), ("model", model)])
            pipe.fit(X_train, y_train)
            train_pred = integerize(pipe.predict(X_
train))
            val_pred = integerize(pipe.predict(X_
val))
            results.append({
                "feature_ratio": ratio,
                "model": name,
                "n_consonant_features": len(cons_
cols),
                    **{f"train_{k}": v for k, v in
scores(y_train, train_pred).items()},
                **{f"val_{k}": v for k, v in scores(y_
val, val_pred).items()}
            })
            predictions.append(pd.DataFrame({
                "feature_ratio": ratio,
                "model": name,
                LANG: val[LANG].values,
                "actual": y_val.values,
                "predicted": val_pred,
                "error": val_pred - y_val.values,
                "absolute_error": np.abs(val_pred -
y_val.values)
            }))
            path = model_dir / f"{ratio.replace('%',
'pct')}_{name.replace(' ', '_')}.joblib"
            joblib.dump(pipe, path)
            saved[(ratio, name)] = path
    results_df = pd.DataFrame(results).sort_
values(["val_MAE", "val_RMSE", "val_R2"],
ascending=[True, True, False]).reset_index
(drop=True)
    predictions_df = pd.concat(predictions,
ignore_index=True)
    best = results_df.iloc[0]
    results_df.to_excel(BASE_DIR / "validation_
model_results.xlsx", index=False)
    predictions_df.to_excel(BASE_DIR /
"validation_predictions.xlsx", index=False)
    joblib.dump(joblib.load(saved[(best
["feature_ratio"], best["model"])]), BASE_DIR /
"best_model.joblib")
    with open(BASE_DIR / "best_model_info.
json", "w", encoding="utf-8") as f:
        json.dump({
            "selected_feature_ratio": str(best
["feature_ratio"]),
            "selected_model": str(best["model"]),
            "n_consonant_features": int(best["n_
consonant_features"]),
            "validation_MAE": float(best["val_
MAE"]),
            "validation_RMSE": float(best["val_
RMSE"]),
            "validation_R2": float(best["val_R2"])
        }, f, ensure_ascii=False, indent=2)
    with zipfile.ZipFile(BASE_DIR / "all_trained
_models.zip", "w", zipfile.ZIP_DEFLATED) as z:
        for path in model_dir.glob("*.joblib"):
            z.write(path, arcname=path.name)
```

(3) Predictive modeling code

```
from pathlib import Path
import numpy as np
import pandas as pd
import joblib
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
BASE_DIR = Path("/content")
TEST_FILE = BASE_DIR / "test_data_15(변수30%).xlsx"
LIGHTGBM_MODEL = BASE_DIR / "best_model.joblib"
RIDGE_MODEL = BASE_DIR / "30pct_Ridge_Regression.joblib"
TARGET_COL = "자음"
LANG_COL = "언어"
test_df = pd.read_excel(TEST_FILE, engine="openpyxl")
test_df.columns = [str(c).strip() for c in test_df.columns]
test_df = test_df.iloc[2:].copy()
test_df[TARGET_COL] = pd.to_numeric(test_df[TARGET_COL], errors="coerce")
test_df = test_df[test_df[TARGET_COL].notna()].copy()
test_df[TARGET_COL] = test_df[TARGET_COL].astype(float)
for col in ["위도", "경도"]:
    test_df[col] = pd.to_numeric(test_df[col].astype(str).str.replace(",", ".", regex=False), errors="coerce")
for col in ["대어족", "area"]:
    test_df[col] = test_df[col].astype("string").fillna("Unknown")
feature_cols = [c for c in test_df.columns if c not in [TARGET_COL, LANG_COL]]
X_test = test_df[feature_cols].copy()
y_test = test_df[TARGET_COL].copy()
lightgbm_model = joblib.load(LIGHTGBM_MODEL)
ridge_model = joblib.load(RIDGE_MODEL)
def integerize(pred):
    return np.clip(np.rint(pred), 0, None).astype(int)
def evaluate(y_true, y_pred):
    return {
        "MAE": mean_absolute_error(y_true, y_pred),
        "RMSE": np.sqrt(mean_squared_error(y_true, y_pred)),
        "R2": r2_score(y_true, y_pred)
    }
lgbm_pred = integerize(lightgbm_model.predict(X_test))
ridge_pred = integerize(ridge_model.predict(X_test))
results_df = pd.DataFrame([
    {"model": "LightGBM", **evaluate(y_test, lgbm_pred)},
    {"model": "Ridge Regression", **evaluate(y_test, ridge_pred)}
])
predictions_df = pd.DataFrame({
    LANG_COL: test_df[LANG_COL].values,
    "actual": y_test.values,
    "LightGBM": lgbm_pred,
    "LightGBM_abs_error": np.abs(lgbm_pred - y_test.values),
    "Ridge": ridge_pred,
    "Ridge_abs_error": np.abs(ridge_pred - y_test.values)
})
with pd.ExcelWriter(BASE_DIR / "test_two_model_results.xlsx", engine="openpyxl") as writer:
    results_df.to_excel(writer, sheet_name="test_summary", index=False)
    predictions_df.to_excel(writer, sheet_name="test_predictions", index=False)
```