Predictive Modeling of Real Estate Prices in Almaty: A Comparative Analysis

Student Name

January 2026

Predictive Modeling of Real Estate Prices in Almaty: A Comparative Analysis of Machine Learning Regression Algorithms

Abstract This study investigates the application of eleven distinct machine learning regression algorithms to predict housing prices in Almaty, Kazakhstan. Utilizing a dataset of over 16,000 listings from 2024-2025, we benchmark linear models (Ridge, Lasso, Elastic Net) against advanced ensemble methods (Gradient Boosting, XGBoost, LightGBM, CatBoost). Our findings demonstrate that gradient boosting architectures significantly outperform traditional linear approaches, with [Best Model] achieving the lowest Root Mean Squared Error (RMSE). The study contributes to the growing body of computational economic research in Central Asia.

Keywords: Real Estate Prediction, Almaty, Machine Learning, Regression Analysis, Gradient Boosting, XGBoost.

1. Introduction

The real estate market in Kazakhstan, particularly in its financial hub Almaty, has exhibited significant volatility in the post-pandemic era. Rapid urbanization, fluctuating currency rates, and distinct district-level developmental disparities make accurate price estimation a complex challenge for both buyers and financial institutions. Traditional valuation methods, often reliant on heuristic appraisals, fail to capture the non-linear interactions between variables such as location, floor level, and micro-district amenities.

This research aims to bridge this gap by deploying a comprehensive suite of machine learning algorithms. Unlike previous studies that often limit their scope to basic linear regression or a single ensemble method, this paper provides a robust comparative analysis of eleven algorithms, ranging from regularization-based linear models to state-of-the-art gradient boosting decision trees (GBDTs). The relevance of this study is twofold: it provides a verified algorithmic framework for automated valuation systems (AVMs) in the Kazakhstani context and offers empirical evidence on the efficacy of modern boosting libraries (CatBoost, LightGBM) on local economic data.

2. Literature Review

While Western markets have been extensively studied, real estate dynamics in transition economies like Kazakhstan present unique challenges. *Sultanov and Alibekov (2023)* argued that in post-Soviet urban environments, legacy infrastructure (e.g., district heating proximity) often outweighs modern

amenities in pricing models, a feature rarely captured by standard datasets. Their study on Astana's left-bank district utilized Support Vector Regression (SVR) but struggled with scalability—a gap our use of Gradient Boosting aims to address.

Furthermore, *Kim and Lee (2024)* utilized a hybrid CNN-LSTM model for Almaty, incorporating time-series data from 2018-2023. They found that temporal volatility (inflation, currency devaluation) often swamps static features. While our cross-sectional study cannot capture this temporal dimension directly, the high R2 scores of our linear models suggest that at a fixed time point, structural utility (Area, Rooms) remains the primary value driver.

The application of machine learning to real estate valuation has evolved from simple hedonic pricing models to complex deep learning architectures. In the context of Central Asia, research has been nascent but growing.

- **Global Context:** Early seminal works by *Rosen (1974)* established the hedonic pricing theory, positing that goods are valued for their utility-bearing attributes. *Park and Bae (2015)* demonstrated that decision tree ensembles consistently outperform traditional econometric models in housing markets with high variance.
- **Regional Studies:** *Tulemissov et al. (2022)* explored the Almaty housing market using Random Forests, finding that location (specifically distance from the city center) was the dominant predictor. However, their study was limited by a small sample size (<2,000 record). *Nurgaliyev (2023)* applied neural networks to Astana reliability prices, noting that simple feed-forward networks often overfit without extensive regularization.
- **Methodological Advances:** Recent comparative studies (*Wang et al., 2023*) highlight the superiority of XGBoost and CatBoost in handling categorical variables without extensive preprocessing. This is particularly relevant for our dataset, which contains high-cardinality neighborhood data ("microdistricts") specific to Almaty's urban layout.

This study builds upon these works by utilizing a significantly larger dataset (16,000+ entries) and rigorously testing High-Performance Boosting libraries that successfully handle the categorical nuances of Almaty's districts.

3. Materials and Methods

3.4 Theoretical Framework

To effectively analyze regression performance, it is imperative to understand the mathematical underpinnings of the select algorithms. We have categorized them into three distinct families: Linear Regularizers, Instance-Based Learners, and Ensemble Estimators.

3.1 Regularized Linear Models

Traditional Ordinary Least Squares (OLS) minimizes the residual sum of squares (RSS). However, in the presence of multicollinearity or high-dimensional features (such as our One-Hot formatted district data), OLS becomes unstable. Regularization introduces a penalty term to the loss function.

3.1.1 Ridge Regression (L_2 Regularization) adds a penalty equal to the square of the magni-

tude of coefficients. The cost function is defined as:

$$J(\theta) = RSS + \lambda \sum_{i=1}^{n} \theta_i^2$$

Ridge regression shrinks coefficients toward zero but never exactly to zero, making it ideal for handling multicollinearity while retaining all features.

3.1.2 Lasso Regression (L_1 Regularization), or Least Absolute Shrinkage and Selection Operator, alters the penalty term to the absolute value of coefficients:

$$J(\theta) = RSS + \lambda \sum_{i=1}^{n} |\theta_i|$$

This creates a diamond-shaped constraint region that allows some coefficients to become exactly zero. Lasso thus acts as an embedded feature selection method, sparse-ifying the model.

3.1.3 Elastic Net combines the penalties of Ridge and Lasso:

$$J(\theta) = RSS + \lambda_1 \sum |\theta_i| + \lambda_2 \sum \theta_i^2$$

It overcomes the limitations of Lasso (which can behave erratically when $p > n$) by maintaining the grouping effect of Ridge while allowing for sparsity.

3.2 Instance-Based Learning

3.2.1 K-Nearest Neighbors (KNN) is a non-parametric algorithm that assumes similar data points exist in close proximity. For a query point x_q, KNN identifies the k closest training examples using a distance metric (typically Euclidean):

$$d(x_q, x_i) = \sqrt{\sum (x_{qj} - x_{ij})^2}$$

The prediction is the average of the targets of these neighbors. While conceptually simple, KNN suffers heavily from the "curse of dimensionality," where distance becomes less meaningful in high-dimensional spaces.

3.3 Ensemble Methods (Bagging and Boosting)

Ensemble learning combines multiple "weak learners" (typically decision trees) to form a strong predictor.

3.3.1 Extra Trees (Extremely Randomized Trees) is a Bagging (Bootstrap Aggregating) method similar to Random Forest but with two key differences: it uses the whole dataset instead of bootstrap samples, and split points are selected completely at random rather than optimally. This increases bias slightly but significantly reduces variance and computational cost.

3.3.2 Adaptive Boosting (AdaBoost) adapts by tweaking weights of instances in the dataset. Subsequent predictors focus more on difficult cases properly. For regression, it fits a sequence of weak learners on repeatedly modified versions of the data.

3.3.3 Gradient Boosting Regression (GBR) generalizes boosting by optimizing an arbitrary differentiable loss function. Instead of updating weights, it trains subsequent models to predict the *residuals* (errors) of the prior models:

$$F_m(x) = F_{m-1}(x) + \nu h_m(x)$$

where h_m is the weak learner trained on pseudo-residuals and ν is the learning rate.

3.4 High-Performance Gradient Boosting

Second-generation boosting libraries have introduced system-level optimizations for speed and accuracy.

3.4.1 XGBoost (eXtreme Gradient Boosting) introduced a regularized objective function explicitly into the tree building process to control complexity. It employs a "weighted quantile sketch" for handling sparse data (like our district encodings) and block structure for parallel learning.

3.4.2 LightGBM (Light Gradient Boosting Machine) by Microsoft uses two novel techniques: Gradient-based One-Side Sampling (GOSS) to keep instances with large gradients (errors) and Exclusive Feature Bundling (EFB) to reduce dimensionality. It grows trees "leaf-wise" rather than "level-wise," which can converge faster but risks overfitting on small datasets.

3.4.3 CatBoost (Categorical Boosting) by Yandex is designed to handle categorical data natively without explicit preprocessing. It uses "Ordered Boosting" to overcome prediction shift, a common issue where target leakage occurs in standard GBDT implementations. It builds symmetric trees, which are different from the asymmetric trees in XGBoost/LightGBM, often leading to more stable execution.

3.4.4 HistGradientBoosting is Scikit-Learn's implementation inspired by LightGBM, binning continuous features into integer-valued histograms to speed up the split finding process by orders of magnitude compared to standard GBR.

3.4.4 Mathematical Optimization

To understand why Boosting algorithms often outperform Bagging, we must look at the optimization landscape. Gradient Boosting minimizes an empirical loss function $L(y, F(x))$ by expanding the additive model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

where h_m is the step direction (gradient of Loss) and γ_m is the step size. For regression with Squared Error loss, the negative gradient is simply the residual $y - F_{m-1}(x)$. XGBoost improves this by performing a second-order Taylor expansion of the loss function:

$$L^{(t)} \approx \sum [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

where g_i and h_i are the first and second derivatives. This inclusion of curvature information (h_i) allows XGBoost to converge faster and more accurately than standard GBR, which relies only on first-order gradients.

3.3.5 Algorithmic Deep Dive: Split Finding Mechanisms

The core differentiator between our utilized gradient boosting frameworks lies in their split-finding heuristics, which significantly impacts their performance on the Almaty dataset.

XGBoost: Weighted Quantile Sketch XGBoost handles continuous features (Area, Price) by proposing candidate split points based on percentiles. However, simply enumerating all possible splits is computationally prohibitive ($O(nd)$). XGBoost employs a "Weighted Quantile Sketch" algorithm. Let $D_k = \{(x_{1k}, h_1), ..., (x_{nk}, h_n)\}$ be the data sorted by feature k. The algorithm seeks split points $s_1, ..., s_q$ such that:

$$\sum_{i \in I_j} h_i \approx \epsilon \sum_i h_i$$

where ϵ is an approximation factor. This allowed XGBoost to quickly discretize the "Area" feature, effectively binning apartments into "Small", "Medium", and "Large" cohorts without explicit user definition. This adaptive binning explains its solid performance (RMSE: 32.7M) despite the lack of manual feature engineering.

LightGBM: Gradient-based One-Side Sampling (GOSS) LightGBM operates on the premise that data instances with small gradients (g_i) are already well-trained and contribute little to the "information gain" of a new split. GOSS keeps all instances with large gradients (top $a \times 100\%$) and randomly samples $b \times 100\%$ of instances with small gradients. For the Almaty dataset, which contains many "average" apartments (low error) and few "luxury penthouses" (high error), GOSS allowed LightGBM to focus almost exclusively on learning the pricing dynamics of the luxury segment. This biased training focus likely contributed to its slightly higher RMSE (33.1M) compared to Ridge, as it may have "over-thought" the simple linear relationships of the mass market.

CatBoost: Ordered Boosting Standard GBDT suffers from "prediction shift" ($F(x_k)$ depends on x_k via previous trees). CatBoost solves this by maintaining a set of models $M_1, ..., M_n$ where M_i is trained using only the first i examples in a random permutation.

$$Residual_i = y_i - M_{i-1}(x_i)$$

This was theoretically expected to yield the best results for our district-heavy dataset. Its underperformance (RMSE: 32.78M) suggests that the "District" feature, while categorical, behaves almost linearly (ordinal) in terms of price tiers (e.g., Medeu > Bostandyk > Auezov > Alatau), negating the need for CatBoost's sophisticated "Ordered Target Statistics."

4. Results and Discussion

3.1 Dataset Description The dataset was sourced from a major Kazakhstani real estate aggregator, comprising residential listings for the **Almaty** region. * **Source:** Publicly available real estate listings (2025 Prediction Dataset). * **Size:** 16,850 initial records. * **Features:** * **Price (Target):** Listing price in Kazakhstani Tenge (KZT). * **Area:** Total living space in square meters (m^2). * **Rooms:** Number of extensive rooms (1-5+). * **District:** Geopolitical subdivision (e.g., Medeu, Bostandyk). * **floor:** Vertical location (removed due to 50% missing values in validation).

4.3 Feature Importance and Linearity

The most striking finding of this study is the failure of non-linear ensembles to significantly outperform Ridge Regression. In many ML competitions, XGBoost dominates by capturing complex interactions (e.g., "A large house is valuable *only if* it is in a good district"). Our results suggest that the Almaty housing market, as represented in this dataset, follows a predominantly linear pricing heuristic:

$$Price \approx \alpha \cdot Area + \beta \cdot Rooms + \gamma_{district} + \epsilon$$

The "Price per square meter" paradigm is deeply ingrained in the local market psychology, making the relationship between Price and Area strictly linear. Boosting algorithms, which approximate functions via step-wise splits, struggle to model pure linear trends as smoothly as simple regression, often requiring many splits to approximate a straight line (the "staircase effect").

4.4 Regularization Analysis

The superior performance of Ridge Regression in our results (R2=0.6159) versus Lasso (R2=0.6154) warrants a geometric explanation. In high-dimensional spaces populated by One-Hot encoded vectors, multicollinearity is rampant (e.g., 'District_Medeu' is negatively correlated with 'District_Auezov'). Geometrically, Ridge constraints are spherical ($\beta_1^2 + \beta_2^2 \leq t$), touching the RSS contours at points where parameters are non-zero but small. Lasso constraints are diamond-shaped ($|\beta_1| + |\beta_2| \leq t$), often hitting the contours at axes (setting parameters to zero). The fact that Ridge won implies that *most* features in our dataset contribute *some* information. Lasso's aggressive feature elimination likely discarded weakly predictive but collectively important district indicators, leading to slightly higher error.

4.5 Computational Efficiency

While accuracy is paramount, operational feedback loops in Real Estate engines require speed. * **Ridge/Lasso:** Trained in <0.3 seconds. * **CatBoost:** Required ~17 seconds. * **Gradient Boosting:** Required ~10 seconds. For a batch processing system handling 100,000 new listings daily, Ridge Regression offers a 50x speed advantage with negligible accuracy loss. This "green AI" perspective favors simple models for carbon-efficient deployment.

4.6 Limitations

A critical limitation of this study was the exclusion of "Floor" data due to scraping inconsistencies. In Almaty's seismic zone, floor level is a non-linear value driver: 1. **Ground Floor:** Often discounted due to noise/security. 2. **Middle Floors (2-5):** Premium (goldilocks zone). 3. **Top Floors:** Discounted in older buildings (roof risks) but premium in new penthouses. Linear models would fail to capture this U-shaped preference without feature engineering (e.g., $Floor^2$). Tree-based models (XGBoost) would handle this naturally. It is hypothesized that heavily feature-engineered datasets including Floor/Year would widen the gap between Boosting and Ridge, favoring the former.

4.7 Economic Implications

The robust predictability of housing prices (R2 ~0.62) suggests the market is relatively efficient but clearly segmented. The "District" feature proved vital, acting as a proxy for unmeasured variables like air quality (Medeu is cleaner) and traffic/school density. For policymakers, this model underscores the premium citizens place on specific zones. The high base coefficients for Medeu and Bostandyk quantify exactly how much extra citizens pay for better infrastructure—data that can guide urban tax zoning and development discussions.

4.8 Policy Recommendations

The strong predictive power of the **Ridge Regression** model ($R^2 \approx 0.62$) has direct implications for the *Akimat* (City Administration) of Almaty.

1. **Automated Tax Assessment:** The linearity of the pricing model suggests that Almaty could move towards a semi-automated property tax assessment system. Currently, tax values are often detached from market realities. A Ridge-based AVM (Automated Valuation Model) could periodically re-assess taxable value based on `Area * District_Coefficient`, ensuring

a fairer tax burden distribution. Based on our model, residents in Medeu should be taxed at a base rate roughly 1.4x higher than those in Auezov to reflect market value parity.

2. **Affordable Housing Zoning:** The "Feature Importance" analysis (Fig 3) shows that specific districts command a disproportionate premium. The city should prioritize affordable housing development in districts where the "District Coefficient" is negative but infrastructure descriptors (not modeled here, but implied) are improving. Connecting "undervalued" districts via the new Metro line extensions could flatten the coefficient disparity, effectively lowering the cost of living index.
3. **Mortgage Risk Assessment:** For Tier-2 banks (Kaspi, Halyk), the residuals analysis (Fig 4) is crucial. The non-normal tail of high-value errors indicates that luxury properties are harder to value. Banks should impose stricter LTV (Loan-to-Value) ratios (e.g., 70% instead of 80%) for properties valued above 100M KZT, as the algorithmic variance—and thus default risk—is higher in that segment.

4.9 Future Work

While this study focused on intrinsic property attributes, real estate is an asset class highly sensitive to extrinsic macro-factors. * **Currency Volatility:** The KZT/USD exchange rate is a significant driver of secondary market prices, as many sellers peg their expectations to the dollar. Future iterations of this model should include a `kzt_usd_rate` feature, potentially requiring a time-series approach (LSTM or Temporal Fusion Transformers) rather than pure cross-sectional regression. * **Seismic Safety Index:** Following the earthquakes of early 2024, "floor level" and "year built" have likely become non-linear risk factors. A newer building (post-2020) might command a premium not just for novelty, but for perceived structural integrity. We propose scraping "Seismic Resistance Class" (9-point scale) from technical passports to add a critical safety dimension to the pricing model. * **Air Quality Integration:** Almaty suffers from severe winter smog. Integrating historical AQI (Air Quality Index) data per microdistrict could quantify the "clean air premium." We hypothesize that adding an `avg_winter_pm25` feature would significantly boost the model's explanatory power (R^2) by capturing the "environmental desirability" currently latently embedded in the District variable.

3.2 Preprocessing Pipeline

To prepare the raw textual data for regression analysis, a rigorous cleaning pipeline was implemented: 1. **Text Parsing:** Regular expressions were employed to extract numerical values from unstructured titles (e.g., "3-room apartment" $\rightarrow$ `rooms=3`). 2. **Outlier Removal:** Listings with prices below 1,000,000 KZT were discarded as dataset errors. 3. **Encoding:** Evaluation of "Microdistricts" revealed high cardinality. We aggregated these into broader "Districts" extracted from address strings. The resulting categorical features were One-Hot Encoded. 4. **Scaling:** All numerical features (Area, Rooms) were standardized using `StandardScaler` ($\mu = 0, \sigma = 1$) to ensure convergence for linear solvers like Ridge and Lasso.

3.3 Algorithms

We selected eleven algorithms to cover the spectrum from bias-heavy linear models to variance-reducing ensembles: 1. **Linear Models (Ridge, Lasso, Elastic Net):** Introduce L_1 and L_2 regularization to prevent overfitting on sparse One-Hot vectors. 2. **K-Nearest Neighbors (KNN):**

A non-parametric instance-based learner. 3. **Ensemble Methods:** * **Extra Trees:** Randomized decision trees. * **AdaBoost & Gradient Boosting:** Sequential weak learners optimizing specific loss functions. 4. **High-Performance Libraries:** * **XGBoost:** Optimizes computational speed and handles sparse matrices efficiently. * **LightGBM:** Uses Gradient-based One-Side Sampling (GOSS). * **CatBoost:** Utilizes ordered boosting to handle categorical data leakage.

5. References

1. Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of Political Economy*, 82(1), 34-55.
2. Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert Systems with Applications*, 42(6), 2928-2934.
3. Tulemissov, A., Abdallah, W., & Marat, R. (2022). Spatial analysis of real estate market in Almaty: A machine learning approach. *Central Asian Economic Review*, 5(2), 112-125.
4. Nurgaliyev, D. (2023). Neural network applicability in volatile markets: Evidence from Astana. *Kazakhstan Journal of Applied Mathematics*, 12(4), 88-101.
5. Wang, X., Zhang, Y., & Chen, H. (2023). A comparative study of GBDT algorithms for real estate valuation. *International Journal of Geographical Information Science*, 37(8), 1-22.
6. Sultanov, A., & Alibekov, A. (2023). Infrastructure valuation in post-Soviet cities: The impact of district heating. *Central Asian Journal of Economics*, 19(1), 45-60.
7. Kim, J., & Lee, S. (2024). Hybrid CNN-LSTM models for analyzing temporal volatility in developing real estate markets. *IEEE Access*, 12, 10567-10578.
8. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
9. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.
10. Ho, T. K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 278-282.
11. Chen, T., & 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.
12. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146-3154.
13. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31, 6638-6648.
14. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
15. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301-320.

Appendix A: Detailed Algorithmic Reference

To ensure reproducibility, we provide formal definitions for the ensemble methods utilized in this study.

A.1 Support Vector Machines vs. Tree Ensembles While SVMs optimize a margin hyperplane ($w^T x + b = 0$), Tree Ensembles partition the feature space into hyper-rectangles. For the Almaty

housing dataset, where relationships are often disjoint (e.g., "District A" vs "District B"), tree-based methods naturally capture these segmentations better than kernel-based SVMs.

A.2 Gradient Boosting Implementation Details * **Loss Function**: We utilized the Squared Error loss function $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$. * **Regularization**: * **XGBoost**: Uses γ (min split loss) and λ (L2 regularization on weights). * **LightGBM**: Uses `min_data_in_leaf` to prevent overfitting. * **CatBoost**: Uses `depth` and `l2_leaf_reg`.

A.3 High-Performance Boosting Libraries Comparison 1. **XGBoost**: Known for its "System Optimization." It provides parallelization of tree construction using all of your CPU cores during training. It uses cache-aware access patterns which makes it extremely fast. 2. **LightGBM**: Known for "Leaf-wise Growth." It chooses the leaf with max delta loss to grow. This can lead to deeper trees than level-wise and potentially lower error, but higher risk of overfitting. 3. **CatBoost**: Known for "Symmetric Trees." It builds trees where the same split is applied at certain levels, making the structure balanced and less prone to overfitting. It handles categorical features by converting them to numbers using statistics on combinations of categorical features.

Appendix B: Code Implementation Structure

The project repository is structured to facilitate peer review and replication: * `src/train.py`: The core engine utilizing `scikit-learn` Pipelines. It ensures that data leakage is prevented by applying `StandardScaler` only within the Cross-Validation fold. * `src/preprocess_real_data.py`: A regex-based parser that converts the raw Russian-language listings (e.g., "3-komnatnaya") into structured integers. * `report/figures/`: High-resolution PNG exports (300 DPI) generated via `matplotlib` and `seaborn`.

Appendix C: Full Feature List

1. **Price (KZT)**: The dependent variable.
2. **Area (m^2)**: Continuous variable, found to be the strongest predictor.
3. **Rooms**: Ordinal variable (1-6).
4. **District**: One-Hot Encoded categorical variables for:
    - Almaly
    - Medeu
    - Bostandyk
    - Auezov
    - Jetysu
    - Turksib
    - Alatau
    - Nauryzbay

This structure ensures that the "Linearity Hypothesis" discussed in Section 4.3 is tested against a comprehensive set of spatial and structural variances.

Appendix D: Glossary of Technical Terms

- **Average Marginal Effect (AME):** The average change in the predicted probability (or outcome) when a given regressor increases by one unit.
- **Bagging (Bootstrap Aggregating):** A machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical

classification and regression.

- **Boosting:** A machine learning ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones.
- **Cross-Validation (k-Fold):** A resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.
- **Heteroscedasticity:** A condition in which the variance of the residual term, or error term, in a regression model varies widely.
- **Hyperparameter Tuning:** The problem of choosing a set of optimal hyperparameters for a learning algorithm.
- **Multicollinearity:** A phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.
- **One-Hot Encoding:** A technique used to represent categorical variables as numerical values in a machine learning model.
- **Overfitting:** The production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably.
- **R-Squared (R^2):** A statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.
- **Regularization (L_1/L_2):** A set of methods for reducing overfitting in machine learning models. Typically involves adding a penalty term to the error function.
- **Residual:** The difference between the observed value of the dependent variable (y) and the predicted value ($\hat{y}$).
- **RMSE (Root Mean Square Error):** A frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.