A Time Independent Gradient Boosting Machine for Rent Prediction in Tokyo 23 Special Wards with Geospatial and Environmental Features

William Steimel

A thesis submitted in partial fulfillment
of the requirements for degree of
Master of Science in
Green Science & Engineering

Department of Information Science
Sophia University
Tokyo, Japan

Contents

A TIME INDEPENDENT GRADIENT BOOSTING MACHINE FOR RENT PREDICTION IN TOKYO 23 SPECIAL WARDS WITH GEOSPATIAL AND ENVIRONMENTAL FEATURES

A PREPRINT

William Steimel
Department of Information Science
Sophia University
Chiyoda, Tokyo 102-8554
steimel65@gmail.com

July 9, 2019

ABSTRACT

Rent prediction and real estate value estimation has long had applications in economics, urban development, and public policy. Recently machine learning techniques have been applied to real estate value estimation in the private sector and competitive machine learning competitions with good results. However, there are some challenges with modeling real estate prices due to the amount of factors that contribute to the overall market which can vary based on geographic region. Tokyo, like any other major city in the world has common but also unique factors that contribute to rent pricing and valuation. This paper introduces an approach that uses Gradient Boosting Decision Tree (GBDT) algorithm LightGBM (LGBM) to predict rent in Tokyo's 23 special wards based on real estate, geospatial, and environmental features. This approach achieves performance gains through the combination of environmental data sources and feature engineering techniques in comparison to a baseline model that only uses rental features. The proposed model achieves price predictions within 5.01 % and 11,200 yen which is an improvement on the baseline model's error of 5.91 % and 12,200 yen. This paper also explores the factors that contribute to rent value in Tokyo's 23 wards based on LightGBM's optimization from an environmental, geospatial, and feature engineering perspective.

***K*eywords** Gradient Boosting · Feature Engineering · LightGBM · Rent Prediction · Tokyo

1 Introduction

Rent is one of the biggest expenditures for households around the world and there is no denying the rental markets' economic impact. According to the OECD, around 22 % of gross adjusted disposable income is spent on housing expenditures in Japan which includes rental cost [1]. In addition, the United Nations reports that currently 55 % of the world's population live in urban areas which is expected to increase to 68 % by 2050 [2]. With expected urban growth and the impact that rent has on the economy and urban development, studies regarding the urban rental market as well as rent prediction models are going to be increasingly useful tools for urban planners, real estate companies, and public policy developers.

Algorithms for real estate price prediction and analysis have recently disrupted traditional pricing methods used in the real estate industry. Some companies in the private sector which have utilized data and machine learning approaches to provide more value to customers include Zillow, GeoPhy, PriceHubble, and Suumo in Japan among many others [3, 4, 5, 6]. Perhaps the most famous example is the Zillow housing index Zestimate, which has been used to accurately estimate the value of a home based on various features and market factors [7]. Zillow also recently held a machine learning competition to improve the base Zestimate algorithm with a grand prize of $ 1,000,000 which shows the price companies are willing to pay to improve these types of models [8]. Estimation of real estate values and factors contributing to overall value has also been a problem frequently explored in economics and public policy literature.

Rent prediction is a challenging problem for a number of reasons including the amount of variables, the variation in each geographic market, temporal aspects like peak and off-peak seasons, as well as the variety of market segments. Machine learning, specifically supervised learning techniques which learn through given labeled training data to generalize on unseen data can be used to further improve on real estate prediction models. This research seeks to build a time independent predictive model that can accurately predict rent in Tokyo through the combination of a rental listings' descriptive features as well as geospatial, environmental, and manually constructed features.

This paper is organized as follows: section 1 will begin with a literature review of associated research. In section 2, the collected data sets used in the models are explained. In section 3, the data cleaning and pre-processing steps are covered. In section 4, experimental methodology is covered. Section 5 reviews the computational results from the experiments. Section 6 wraps up this paper with future work and a conclusion.

1.1 Related Works

This section refers to related works for rental and housing market valuation, including traditional econometric approaches, recent machine learning approaches, and Tokyo real estate market specific research.

1.1.1 Traditional Approaches

Due to its roots in econometrics, the most traditional approach to rental and housing market estimation is the Hedonic regression model. The Hedonic pricing model was first proposed in Sherwin Rosen's paper "Hedonic Pricing and Implicit Markets: Product Differentiation in Pure Competition" and is used often in housing research for understanding of characteristics that contribute to price [9]. It states that a product is a differential commodity whose value is determined as a function of its various characteristics. In the case of the housing market, these characteristics include building related qualities like number of rooms, area, and floor among others. This can be denoted as the following equation, which is the same equation as a Linear Regression model:

$$y_i = f(x_i) + \epsilon_i \tag{1}$$

In this equation, $f(\cdot)$ is the Hedonic regression function that determines the average value of y (price) given x (characteristics), while ϵ is the error term.

A key difference between econometric and machine learning applications for the real estate market is that the former is primarily concerned with parameter estimation, or how variables x and y are related while machine learning applications want to produce predictions y based on input features x [10]. Perhaps one of the biggest benefits of a Hedonic estimation model is its interpretability although it may not have the predictive performance that some state of the art machine learning methods have. Hedonic regression modeling is still commonly used for real estate research and has been used in recent research papers applied to housing markets like Singapore and Turkey [11, 12]. Although Hedonic modeling traditionally refers to a linear model there are many different techniques that can be applied which has also recently expanded to include the use of machine learning methods [13, 14, 15].

1.1.2 Machine Learning Approaches

Machine learning approaches modeled as a supervised learning regression problem have been applied with good results for real estate modeling in comparison to more traditional linear methods.

One study tested several machine learning algorithms on high end real estate assets in the Salamanca district in Madrid, Spain [16]. The experiments compared the advantages and weaknesses of models including regression trees, k-nearest neighbors, support vector machines, and neural networks. The features used to model price were standard real estate features and included the following: Zone, Postal code, Street name, Street number, Floor number, Type of asset, Constructed area, Floor area, Construction year, Number of rooms, Number of baths, is penthouse, is duplex, has lift, has box room, has swimming pool, has garden, has parking, parking price, and community costs. The results concluded that ensembles of regression trees performed the best and that more complex machine learning algorithms were able to outperform traditional linear or hedonic regression models in prediction accuracy.

An additional predictive model for housing prices in Fairfax county, Virginia was constructed and tested with machine learning techniques [17]. The study tested four different machine learning algorithms including C4.5, RIPPER, naive bayesian, and AdaBoost and concluded that the RIPPER model performed the best.

Machine learning was also employed to predict house prices with the Kaggle Melbourne Australia housing market dataset [18]. Multiple machine learning methods including linear regression, polynomial regression, regression trees, neural networks, along with stepwise feature selection with SVM, and PCA with SVM were tested to see which method

would perform best. The stepwise feature selection applied before support vector machines was concluded to be most effective although slower than the other methods.

Ensemble techniques were also exclusively tested on rental data in Bangladesh [19]. Four ensemble methods including ensemble bagging, ensemble AdaBoosting, ensemble gradient boosting, and ensemble XGboost were tested by stacking them on base predictors models including linear regression, ridge regression, lasso regression, elastic net regression, random forest, neural networks, and support vector machines. These models were used to predict rent on a dataset with 3,505 houses and 19 features and found that ensemble gradient boosting performed the best with ensemble AdaBoosting at the worst.

Although many research papers focus on comparison of algorithms, this research would like to take a different approach and analyze the addition of environmental features as well as feature engineering techniques. In addition, XGBoost was utilized in studies where Gradient Boosted Decision Tree models were used but this research would like to experiment with LightGBM which is a recent competitor.

1.1.3 Tokyo Real Estate Market

The Tokyo housing market has its own factors that contribute to market price and this section reviews different research in economics applied to understanding the real estate market in Tokyo.

One paper utilized traditional real estate estimation models like the hedonic model, geostatistics based regression kriging, and spatial autoregressive error model from spatial econometrics on the Tokyo 23 wards apartment market [20]. It was found that the regression kriging and spatial autoregressive were found to be advantageous over traditional hedonic regression in predicting rent. The model used general apartment descriptive features like bus time, walk time, log of floor area, age of unit, reinforced concrete, nos. of rooms, one-room type, 1K-type, parking lot, self-locking, and dummy variables for wards.

In addition, Tokyo is considered to be prone to natural disasters with risk of earthquakes and flooding near most river areas. One study analyzed the relationship between earthquake risk and rental prices through the use of hazard maps and found that rents were significantly lower in more risky areas [21]. This contribution makes environmental factors like earthquake and building risk possibly useful features for building a rent prediction model in Tokyo.

Amenities are something that we all are conscious of when looking for a new neighborhood to live in. A Hedonic approach was used to classify amenities into 24 groups to study the relationship between urban amenities and rent in the Tokyo metropolitan area [22]. A key finding was that restaurants and educational facilities contributed to higher rents for an area while cemeteries and video arcades had the opposite effect.

Tokyo is well known for its excellent public transportation infrastructure with line and station being a key factor for most apartment hunters. In regards to station, locations, and rent, a spatial hedonic model was applied to understand how proximity to train and subways impact rent values and found that at least the first three stations should be considered to have significant impact on rent value [23]. This will also be important to consider during this study.

To my knowledge, so far none of these studies have applied state of the art machine learning techniques to predict rent in the context of Tokyo. In addition, none have applied the use of feature engineering techniques or environmental factors in prediction other than earthquake risk data which is why this research seeks to explore these avenues further.

1.2 Environmental Features and Feature Engineering

Machine learning model performance is heavily impacted by the features that are used as inputs. This research wishes to test geospatial and environmental features and their effectiveness towards rent estimation. Environmental and geospatial information are secondary factors in comparison to features that describe the actual rental unit or building. Tokyo has recently emphasized the development of a sustainable city with its environmental policy "Creating a Sustainable City" [24]. With Tokyo's ambitious environmental plan, it is interesting to examine whether these environmental factors can contribute to overall rental market value and real estate estimation. Key environmental data points like air quality, disaster risk, parks, crime, and land classification among others will be tested as features on whether they contribute to more accurate rent prediction within Tokyo's 23 special wards. The impact of these environmental factors as well as their effectiveness is analyzed through final model performance.

This study also presents a number of engineered features and techniques that can be used in other real estate estimation algorithms. Feature engineering is the process of generating features through the transformation of variables to augment predictive accuracy. Although feature engineering contributes so much to performance, it is often seen as a black art as it often requires domain specific knowledge and human intuition [25]. In this research, feature engineering methods will be applied against real estate, geospatial, and environmental data to build the optimal rent estimation model.

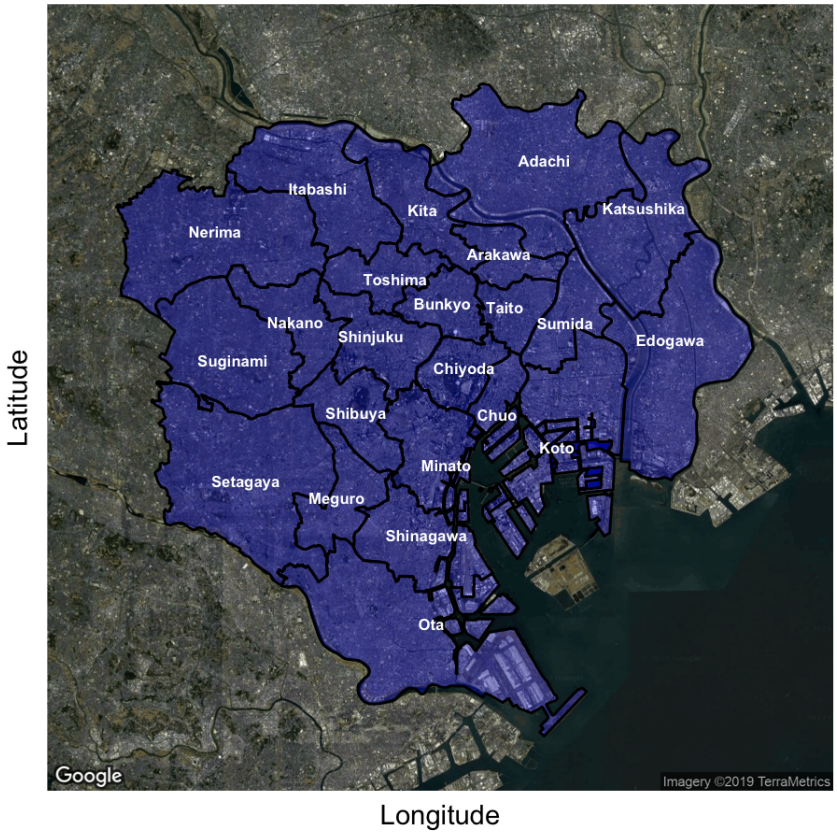


Figure 1: Tokyo 23 Special Wards

2 Datasets

This section details the data collected and used to build a rent prediction model for Tokyo's 23 special wards. To create a data set for each model, a combination of rental property data, geospatial data, and multiple environmental sources were combined from various online sources.

2.1 Suumo (Base Dataset)

This research uses rental data from real estate platform Suumo (Recruit) which is an online platform in Japan for housing, real estate buying and selling, and rental support information [6]. The data was extracted from the platform for rental units within the Tokyo 23 special wards on March 27, 2019. The data consists of rental properties belonging to 2,965 districts in the 23 special wards. The complete data set contains 165,983 listings and 15 features that describe each rental unit. A translated feature dictionary and a map of the study area can be viewed in table 1 and figure 1 respectively. This data can be referred to as the base dataset as all other data sets will be joined to it and its features will contribute the most impact towards the rent prediction model.

2.2 Google Cloud API's

To generate geospatial features, Google Maps Geocoding was utilized to collect the longitude and latitude for each administrative district and closest three stations in the Suumo rental data [26]. The Distance Matrix API was also used to calculate the distance and time between these stations and the major hub stations of Tokyo, Shinjuku, Shibuya, Ueno, Ikebukuro, and Shinagawa [27]. The times to each station are based on the Google Maps calculation of travel time at 8:00 AM on a weekday. These six stations were chosen as people living in different regions of Tokyo may utilize different hub stations and have different preferences. This may provide further context into station preferences between each region in Tokyo. A map containing the locations of these hub stations can be found in figure 2 as well as feature dictionaries in table 2 and 3. The data was joined to the base Suumo dataset by the administrative district (location) and station.

Table 1: Suumo (base) dataset

Feature	Description
title	Suumo listing description
Ku	Tokyo Ku that the unit is located in
location	Administrative district (Also known as chome)
station_1	Closest station and travel time
station_2	Second closest station and travel time
station_3	Third closest station and travel time
yrs	Age of the building in years
heights	Height/depth of the building
floor	Floor the rental unit is on
rent	Monthly rent
admin	Monthly administrative Fee
deposit	Deposit fee when starting a contract
gratuity	Gratuity fee when starting a contract
floor_plan	Room layout of rental unit
area	Total space in meters squared

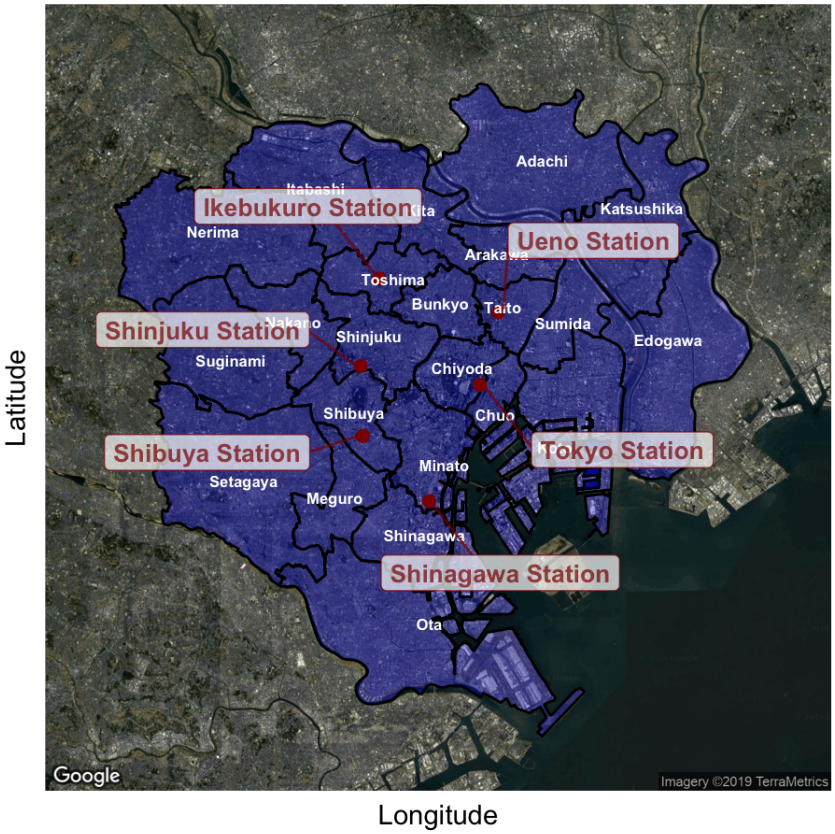


Figure 2: Tokyo Hub Stations

Table 2: Google Maps Geocoding Data (Location)

Feature	Description
location	Administrative district (link to base data on location)
LAT	Latitude
LON	Longitude

Table 3: Google Maps Distance Matrix Data

Feature	Description
Stations	Station name (link to base data on Station_1)
LAT	Latitude
LON	Longitude
ShinjukuMade	Time to Shinjuku
ShinjukuDist	Distance to Shinjuku (km)
TokyoMade	Time to Tokyo
TokyoDist	Distance to Tokyo (km)
ShibuyaMade	Time to Shibuya
ShibuyaDist	Distance to Shibuya (km)
IkebukuroMade	Time to Ikebukuro
IkebukuroDist	Distance to Ikebukuro (km)
UenoMade	Time to Ueno
UenoDist	Distance to Ueno (km)
ShinagawaMade	Time to Shinagawa
ShinagawaDist	Distance to Shinagawa (km)

2.3 Tokyo Regional Earthquake Risk Survey

The Tokyo Regional Earthquake Risk Survey conducted by the Tokyo Bureau of Urban Development details the risk posed to each administrative district in the event of an earthquake [28]. Tokyo surveys earthquake risk in 5,177 administrative districts to support planning and disaster resilience. Data was collected from all administrative districts within Tokyo's 23 special wards and features were translated from Japanese to English. Data measured by this survey include scores, rankings, and defined risk levels for fire, building, livelihood, and overall risk during the event of an earthquake. Land classification is also a useful feature that represents the physical geography characteristics of the administrative district which may be utilized for further risk modeling and feature engineering. The data was merged with the base Suumo dataset by administrative district (location) and can be referenced in table 4.

Table 4: Tokyo Regional Earthquake Survey

Feature	Description
Land Classification	Land classification by administrative district
Building_Risk	Risk of building destruction
Building_Risk_Rank	Ranking in dataset for building destruction risk
Building_Risk_Level	Level of risk to buildings which is evaluated relative to other districts
Fire_Risk	Risk of fire
Fire_Risk_Rank	Ranking in dataset for fire risk
Fire_Risk_Level	Level of risk to fire which is evaluated relative to other districts
Life_Difficulty_Risk	Risk to livelihood in event of earthquake based on access to everyday services.
Life_Difficulty_Risk_Rank	Ranking in dataset for livelihood risk
Life_Difficulty_Risk_Level	Level of risk to livelihood which is evaluated relative to other districts
Total_Risk	Overall risk
Total_Risk_Rank	Overall risk ranking in dataset
Total_Risk_Level	Level of overall risk which is evaluated relative to other districts

2.4 Tokyo Air Quality

Air quality is often overlooked when looking for a new apartment or mansion but possibly has some indirect impact on rent valuation. Time series data was collected from the Japan Ministry of the Environment's Atmospheric Environmental Regional Observation System: AEROS from the period of April 2018 - April 2019 [29]. This system captures the current state of air pollution round the clock, at 10 minute intervals at various sensor locations around the country. Air quality indicators for pollution like SO2(ppm), NO(ppm), NO2(ppm), Ox(ppm), SPM(mg/m3), PM2.5(ug/m3), wind direction(16Dir), wind speed(m/s), temperature, and humidity were gathered from 33 different sensors around the Tokyo region. The features were aggregated by their minimum, mean, median, maximum, and standard deviation values for the entire year and used in the predictive model. Since sensors do not collect all types of readings, any sensor

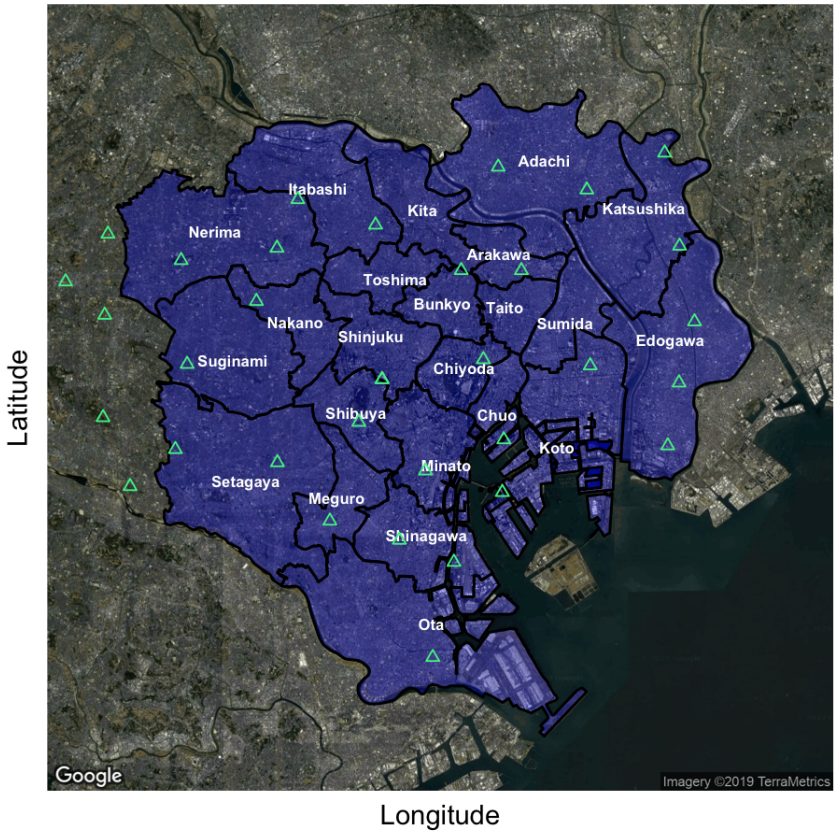


Figure 3: Tokyo Air Quality Sensors

Table 5: Tokyo Air Quality

Feature	Description	Aggregates (1yr)	Total Features
SO2 (ppm)	Sulfur Dioxide	min, median, mean, max, std	5
NO (ppm)	Nitrogen Monoxide	min, median, mean, max, std	5
NO2 (ppm)	Nitrogen Dioxide	min, median, mean, max, std	5
Ox (ppm)	Photochemical Oxidant	min, median, mean, max, std	5
NMHC (ppmC)	Non-Methane Hydrocarbons	min, median, mean, max, std	5
SPM (mg/m3)	Suspended Particulate Matter	min, median, mean, max, std	5
PM2.5 (ug/m3)	Fine Particulate Matter	min, median, mean, max, std	5
WS (m/s)	Wind Speed	min, median, mean, max, std	5
TEMP	Temperature	min, median, mean, max, std	5
HUM	Humidity	min, median, mean, max, std	5
WD	Wind Direction Counts by direction	count by 18 wind directions	18

with null values was imputed with its closest sensor via euclidean distance. For any closest sensor with null readings the next closest sensor was used. These sensors were then joined with the base dataset by calculating the closest sensor to each administrative district (location) in the dataset. A feature dictionary is provided in table 5 and locations of all air quality sensors in or near the study area can be found in Figure 3.

2.5 Tokyo Parks

Parks in Tokyo are considered essential facets to communities due to their use for social gathering, air quality, and as evacuation sites during times of disaster. Parks data was collected from the Tokyo Bureau of Construction for each Ku [30]. The dataset contains descriptive statistics for the various types of parks based on Tokyo's classifications including

Table 6: Tokyo Parks

Category	Feature Group	Aggregates	Total Features
General	Area	Total area for Ku (hectares)	3
	Population	Total population for Ku	
	Population Density	Total area per person	
Urban Parks	Metropolitan Parks	Totals, area (hectares)	10
	Municipal Parks	Totals, area (hectares)	
	National Government Parks	Totals, area (hectares)	
	Urban Parks (subtotal)	Totals, area (hectares)	
	Urban Parks (ratio)	Area per person, park to area ratio	
Non-Urban Parks	Seaside Parks	Totals, area (hectares)	8
	Municipal Parks	Totals, area (hectares)	
	Non-Urban Parks (subtotal)	Totals, area (hectares)	
	Non-Urban Parks (ratio)	Area per person, park to area ratio	
Municipal Parks	Public Parks (subtotal)	Totals, area (hectares)	4
	Public Parks (ratio)	Area per person, park to area ratio	
Other Parks	National Parks	Totals, area (hectares)	4
	Private Parks	Totals, area (hectares)	
Totals	Total (subtotal)	Totals, area (hectares)	4
	Total (ratio)	Area per person, park to area ratio	

urban parks, non-urban parks, municipal parks, and their sub-classifications by Ku. The data was joined to the base DataFrame by Ku and the features can be found in table 6 grouped by category.

2.6 Tokyo Crime

Aggregated crime statistics for 2018 (Heisei 30) were collected for each administrative district in Tokyo from the Tokyo Metropolitan Police Department website [31]. Features were translated from Japanese to English and contain totals for felonious crimes, violent crime, burglary and larceny, non-intrusive larceny, and other categories as well as breakdowns for respective subcategories of crime. The data was linked at the administrative district (location) level and is detailed in table 7 by category.

2.7 Land Use by District

Land use by district data was collected from Tokyo Statistical Handbook 2017 and the Tokyo Bureau of Urban Development for year 2016, Heisei 28 [32]. The dataset details how land is used in each Ku including how much area is classified as residential, parks, unused, roads, farms, waterbodies, forest, fields, as well as other-use designated land in hectares. The feature dictionary can be found in table 8.

2.8 Pollution Complaints

The complaints about pollution dataset was also collected from the Tokyo Statistical Handbook 2017 and provided by the Tokyo Ministry of the Environment. It contains aggregate totals of complaints of various types of pollution by each Ku [32]. The features used from this dataset can be found in table 9.

2.9 Challenges

Challenges with the previously mentioned datasets observed during this research are detailed further in this section.

- **Outliers -** There are many outliers in the dataset especially regarding the total rent in each Ku. The target distribution is a long tail distribution. There is a clear disconnect between average rental properties and luxury rental properties which makes it more difficult to generalize on rent value. A model robust to outliers will need to be developed to account for this challenge. Tree based methods are said to be more robust to outliers in comparison to linear models which is why they were selected for this study.

- **Dirty Data-** The data was extracted from online and the rental listings seem to be inputted by an end user (within the constraints of Suumo). Data is also collected from multiple decentralized sources and needs to be

Table 7: Tokyo Crime

Category	Features	Description	Total Features
Felonious Crime	Robbery	Total robbery	3
	Other	Total for other felonies	
	Total	Total all felonies	
Violent Crime	Rape	Total rapes	5
	Assault	Total assaults	
	Threat	Total threats	
	Blackmail	Total blackmail	
	Total	Total violent crime	
Burglary and Larceny	Safe_Theft	Total safe thefts (including banks)	9
	School	Total thefts at school	
	Office	Total thefts at office	
	FoodStall	Total thefts at food stall	
	House	Total thefts at house	
	burglary	Total break and enter thefts	
	Abandoned_house	Total thefts at abandoned property	
	Other	Total other burglary and larceny	
	Total	Total for burglary and larceny	
Non-Intrusive Larceny	Bike	Total bicycle theft	11
	Motorcycle	Total motorcycle theft	
	Car	Total motor vehicle theft	
	Vending_Machine	Total vending machine theft	
	Construction_Site	Total theft at construction sites	
	Pickpocket	Total pickpockets	
	purse_snatching	Total purse thefts	
	bag	Total bag thefts	
	shoplifting	Total shoplifting incidents	
	other	Total for other larceny	
	Total	Total non-intrusive larceny	
Other Crimes	Fraud	Total frauds	6
	Embezzlement	Total embezzlement	
	Intellectual Crime	Total intellectual crimes	
	Gambling	Total gambling offense	
	Other Offenses	Total other offenses	
	Total	Total other crimes	
Total	Total_Crimes	Aggregate total of crimes	1

Table 8: Tokyo Land Use by District

Feature	Description (Area in hectares)
Total	Total area
Residential	Residential Areas
OtherUse	Quarries, refuse dumping-grounds.
OtherUse(Open)	Storage space, parking lots, exhibition space, construction camps.
Parks	Parks, athletic fields, baseball grounds
Unused	Residential sites before construction, demolished sites, deserted buildings, reclaimed land
Roads	Urban roads, pavements, bicycle roads
roads2	Roads, railway tracks, monorail tracks, airports, seaports
Farm	Rice paddies, ordinary fields, orchards
Water	Rivers, canals, lakes, ponds
WoodsForest	Woodlands, bamboo groves
Fields	Grasslands and other uncultivated land

Table 9: Tokyo Pollution Complaints

Feature	Description
Total_Complaints	Total aggregate complaints
Air_Pollution	Air pollution related complaints
Water_Pollution	Water pollution related complaints
Soil_Pollution	Soil pollution related complaints
Noise	Noise complaints
Vibration	Vibration complaints
ground_subsidence	Ground subsidence complaints
Offensive_odors	Offensive odor complaints
other_pollution	Other complaints

merged. There is no consistent naming convention for administrative districts within each dataset which makes proper data pre-processing essential so that the datasets can properly be merged. An additional challenge is the kanji symbols in numeric columns which will require different text pre-processing depending on the character. Features with characters like these will need to be cleaned up before modeling.

- **Administration Fee -** Administration fee is a monthly charge that is often considered part of the monthly cost of living in Tokyo. Not every person renting an apartment pays this fee and there seems to be no trend on which apartments have this fee. This feature adds a great deal of noise to the data as their is no consistent pattern in administrative fees for apartments.
- **High Categorical Feature Cardinality –** Categorical features like location, Ku, stations, and train lines have high cardinality and when encoded pose a challenge for the data dimensionality and computational cost. Methods to reduce dimensions, feature reduction, or feature engineering methods will need to be explored.
- **Japanese/English Data –** Most of the data required for these experiments requires proficiency in Japanese to collect and evaluate. Much of the data will need to be translated to English for research purposes.

3 Data Cleaning/Pre-Processing

This section details the data cleaning and pre-processing steps necessary before modeling can be performed on all datasets. The Pandas, Scikit-Learn, and NumPy libraries were used to perform all pre-processing operations [33, 34, 35].

3.1 Outlier Removal

Multiple strategies for removing outliers were employed which will be outlined below. 351 samples with a target value above 800,000 or below 23,000 were removed from the dataset as they were on the upper and lower extremes of the data. In addition, rent per meter was calculated for each property. Data was then grouped by Ku where any sample with rent per meter 5x above or below the average for its age was removed. Due to less samples for older apartments, any apartments from age 50-98 were grouped into bins of 10 years. 15 samples were removed with this technique. In addition, 20 apartments coded as 99 years old were removed as the large spike in values at this age seemed to indicate the value was being used erroneously as a placeholder for unknown values. Lastly, 2 properties with unbelievable room layouts were removed as these examples were just not representative of the entire dataset. In total, 388 data points were removed with 165,696 samples remaining in the entire dataset. Although it is likely more values could have been removed, the goal of this research involves modeling the actual conditions of the rental market which requires being able to deal with the various market segments. After outlier removal figure 4 details the total rental listings per Ku.

3.2 Target Value (Rent + Administration Fee)

The target which is the value that this supervised learning problem aims to predict is created by adding the total monthly rent (yachin) together with the administrative fee (kanrihi). Although not all residents in Tokyo pay an administrative fee, this fee is also paid with rent and can be considered to be a part of the monthly cost for rental. The administrative fee adds a degree of noise to the data as these fees can vary widely for property which presents an interesting and realistic opportunity for modeling rent in Tokyo. The distributions of the target value by Ku for the entire dataset after outlier transformation can be seen in figure 5.

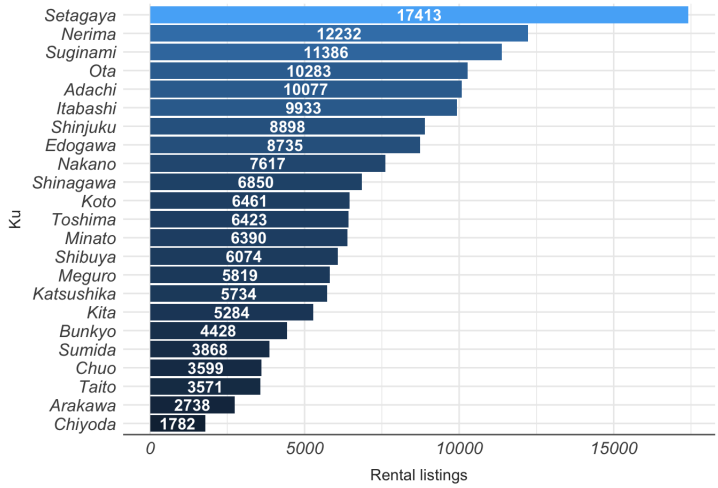


Figure 4: Total Listings by Ku after outlier removal

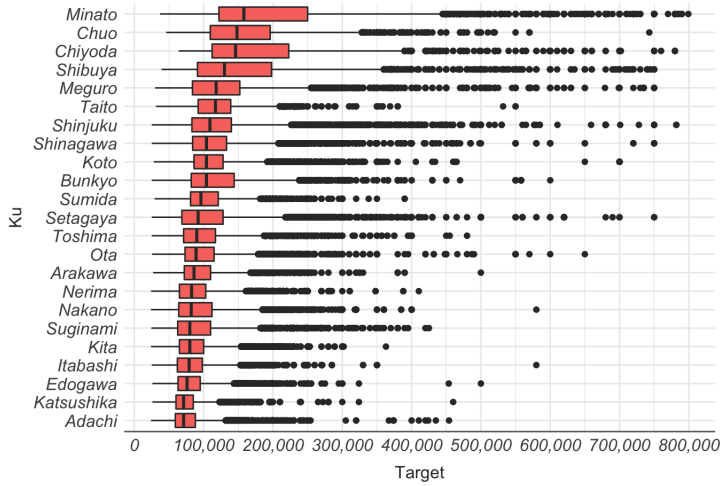


Figure 5: Price distributions of target value by Ku

Table 10: Total Samples per Dataset and Target Descriptive Statistics

dataset	count	mean	std	min	25%	50%	75%	max
train	105,980	108,596.47	65,599.13	24,000	70,000	91,000	124,500	799,310
validation	26,496	108,593.61	65,337.07	24,500	70,000	91,000	125,000	782,000
test	33,119	108,555.13	65,454.43	24,000	70,000	91,000	125,000	790,000

3.3 Train/Validation/Test Split

When working on machine learning problems, one of the first things that should be done with a dataset is splitting of the dataset into train, validation, and test partitions. To preserve the data distribution, the target value was split into 15 quantiles or bins which were used to perform stratified splitting of the dataset. The dataset was first split 80/20 to create a training and test set. The training set was then split 80/20 with the minority set becoming the validation set. The target distribution as well as the descriptive statistics for the Train, Validation, and Test set post-transformation can be found in figure 6 as well as table 10. Due to the long tail of the distribution, the distributions are visualized at a logarithm scale.

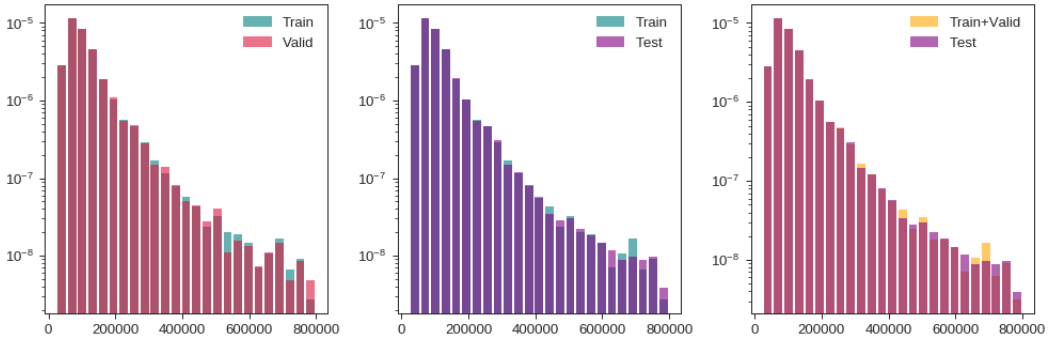


Figure 6: Distributions for Train, Validation, and Test sets (logarithm scale)

4 Methodology

This section details the methodology including the performance metrics, experimental setup, pre-processing steps, algorithm, and hyperparameter optimization used to conduct experiments.

4.1 Performance Evaluation Metrics

Two metrics are used to evaluate experiments in this research paper including the Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). The Root Mean Squared Error is (RMSE) defined as:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

The RMSE was chosen as it gives us the approximate error between actual values and predictions in yen value. From a business perspective, this metric gives us the real value of rent that the predictions are off by.

The second metric, Mean Absolute Percentage Error (MAPE) is defined as:

$$MAPE = \frac{1}{n}\Sigma_{i=1}^{n}|\frac{(y_i - \hat{y}_i)}{y_i}| \times 100 \tag{3}$$

The MAPE metric gives us the approximate percentage of error from the actual prediction. This metric is useful considering that rental prices can range greatly in price depending on rental market segment. The RMSE is more sensitive to higher priced properties which the MAPE can help with evaluation of. These two metrics will also be calculated for each Ku and price bin used for stratified splitting to understand how well the model can generalize for different areas and different price segments.

The Mean Squared Error (MSE), also known as L2 Loss is used as the objective function for optimization as it is the default setting for LightGBM. The MSE is the sum of squared errors between the predictions and actual values.

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

A separate evaluation metric which is the Root Mean Squared Logarithmic Error (RMSLE) is used in the early stopping process to help guide the model training. The RMSLE is similar to the RMSE but also transforms the predicted and actual value by log1p which is defined below:

$$RMSLE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}log(\hat{y}_i + 1) - log(y_i + 1)^2} \tag{5}$$

This evaluation function was chosen due to its robustness to outliers and due to the fact that the target distribution is a long-tail distribution.

4.2 Experimental Setup

The computational environment used for these experiments was the Google Colaboratory cloud based Jupyter notebook environment which utilizes a GPU Tesla T4 with 12.6 GB of RAM and 320 GB of disk [36] .

As mentioned previously, the evaluation metrics used for this study are the RMSE and MAPE score which are compared for three different experiments. The aim of this study is to build a sequentially better model through the addition of geospatial, environmental, and engineered features. These experiments as well as the features used are detailed as the following:

- **Experiment 1: Baseline Model**
  - Suumo base features only
- **Experiment 2: Geospatial and Environmental Model (Geo-Env)**
  - Suumo base features
  - Geospatial and environmental features
- **Experiment 3: Feature Engineering Model (FE)**
  - Suumo base features
  - Geospatial and environmental features
  - Feature engineering

Each model was evaluated based on the average of 5-fold cross-validation conducted 5 times with different random seeds on the combined training and validation set with one final evaluation on the test set. K-fold cross validation is a model-evaluation technique that splits a dataset into folds where $k-1$ folds are used for training and the remaining data is used to generate a performance estimate. This process is done for k folds which results in 5 performance estimates of validation error per cross-validation. These results are then averaged to get a more robust estimate of model generalization performance.

4.3 Model Pre-processing

Data pre-processing is a crucial step for creating a machine learning model that can generalize well. Numerous pre-processing operations were performed which will be detailed by model in the following subsections.

4.3.1 Data Pre-Processing - Baseline Model

Kanji and Text Cleanup

The Japanese language has three alphabets including kanji, hiragana, and katakana which often show up in Japanese data. These symbols or characters come with important meanings about the context of the data but need to be removed to create numeric features before modeling can occur.

In this rental data, the most common characters grouped by the features they appear in are listed below:

- **gratuity, deposit, rent, admin**
  - 万円 - (manen) suffix that indicates cost in 10,000 yens. example: 7万円 - 70,000 yen
- **yrs**
  - 新築 - (shinchiku) Indicates that a building is new.
  - 年 - (nen) A suffix that refers to the age of a building. example: 7年 - 7 years old
- **height, floor**
  - 階 - (kai) Indicates what floor a rental unit is on. example: 5階 - 5th floor
  - 地下 - (chika) - Basement floors in a building. example: 2地下 - 2 basement floors
  - 地上 - (chijyou) - Floors above ground in a building. example: 10地上 - 10 floors above ground
  - 建物 - (tatemono) - just indicates building with no special meaning.
  - 平屋 - (hiraya) - one story house
- **Station_1, Station_2, Station_3**
  - 歩 - (ho) - Suffix for walk time to a station. example: 4歩 - 4 mins walking to station
  - 車 - (sha) - Suffix for car travel time to a station. example: 10車 - 10 mins by car to station
  - バス - (basu) - Suffix for bus travel time to a station. example: 15バス - 15 mins by bus to station

Table 11: Baseline Model Features

Feature	Type	Pre-Processing	Definition
yrs	Numeric	Removed Kanji	Age in years of apartment
floor	Numeric	Removed Kanji, imputed highest floor value for multi-floor apartments	Floor of rental unit
deposit	Numeric	Binarized	1 if deposit fee
gratuity	Numeric	Binarized	1 if gratuity fee
area	Numeric	Removed meters squared, log transformed	Area in meters squared after log transform
admin_flag	Numeric	Binarized	1 if admin fee
new	Numeric	Binarized	1 if apartment is new
Basement_Depth	Numeric	Basement value inferred from Kanji in height feature	Building basement depth
Height	Numeric	Total height of building inferred from Kanji in height feature	Building height
entirehouse	Numeric	Binarized	1 if rental is entire house
mezanine	Numeric	Binarized	1 if rental is on mezanine floor
multifloor	Numeric	Binarized	1 if rental has multiple floors
basementdweller	Numeric	Binarized	1 if living in basement
Car_1	Numeric	Extracted from Station_1 Kanji	Car travel time to closest station
Walk_1	Numeric	Extracted from Station_1 Kanji	Walk travel time to closest station
Bus_1	Numeric	Extracted from Station_1 Kanji	Bus travel time to closest station
Car_2	Numeric	Extracted from Station_2 Kanji	Car travel time to second closest station
Walk_2	Numeric	Extracted from Station_2 Kanji	Walk travel time to second closest station
Bus_2	Numeric	Extracted from Station_2 Kanji	Bus travel time to second closest station
Car_3	Numeric	Extracted from Station_3 Kanji	Car travel time to third closest station
Walk_3	Numeric	Extracted from Station_3 Kanji	Walk travel time to third closest station
Bus_3	Numeric	Extracted from Station_3 Kanji	Bus travel time to third closest station
location	Categorical	Text cleanup	Administrative district (chome)
floor_plan	Categorical	Text cleanup	Room layout
Ku	Categorical	None	Ku
Line_1	Categorical	Split from Station_1	Closest station's line
Station_1	Categorical	Split from Station_1	Closest station
Line_2	Categorical	Split from Station_2	Second closest station's line
Station_2	Categorical	Split from Station_2	Second closest station
Line_3	Categorical	Split from Station_3	Third closest station's line
Station_3	Categorical	Split from Station_3	Third closest station

Depending on the kanji suffix and preceding number, we can infer specific features related to the rental property like its rent, age, height, and travel time to station among many others. All of the above text characters were removed to create numeric features. The 万円 (manen) suffixes were removed from the admin fee, rent, gratuity, and deposit fields and multiplied by 10,000 to create monetary values. The 新築 (shinchiku) character was used to create a new binary feature titled new indicating whether a building is new or not. The 年 (Nen) character was stripped from the yrs column to create a numeric building age feature. The 階 (kai) character was removed from the floor feature to create a numeric representation of the floor.

The height feature was split into two columns including basement_depth and height depending on the number prefix of the 地下 (chika) and 地上 (chijyou) characters. Any value with the 平屋 (hiraya) value received its own feature called entirehouse. The 歩 (ho), 車(sha), and バス (basu) suffixes were used to split Station_1, Station_2, and Station_3 into three separate columns each. These features were used to represent times of travel by walking, driving, and bus to the first three stations.

Feature Transformation

Many features were modified to promote optimal modeling in the experiments which are detailed in this section. Basic text cleanup for inconsistent values was performed on all features.

Multiple binary features were created based on existing features. The deposit, admin, and gratuity fields were converted to binary features indicating whether an apartment required a deposit, gratuity, or administration fee. This was done because deposit and gratuity fee often come at the same cost as rent which would be a form of data leakage. The administration fee is also a part of the target value which makes it somewhat correlated with the target. A feature titled multifloor was created because some apartments actually had multiple floors in their data. Any sample with multiple floors was encoded with the highest floor in the floor feature. In addition, a minority of people lived in a basement or mezanine floor so binary features were created to represent these two groups.

The area feature was transformed with the logarithm function to compress the long tail skew of the distribution. Categorical features were left mostly untouched other than some minor inconsistent text cleanup. The final features used in the baseline model after pre-processing can be found in table 11 along with their pre-processing steps.

Table 12: Geospatial/Environmental Model Datasets

Dataset	Source	Description	Join Key
Suumo	Suumo	Apartment and mansion listings for 23 Ku's in Tokyo	location/Ku
Google Cloud API's	Google Cloud	Geospatial coordinates, distance and time calculations to hub stations	location
Tokyo Regional Earthquake Risk Survey	Tokyo Bureau of Urban Development	Earthquake risk by administrative district in Tokyo (chome)	location
Tokyo Air Quality	Tokyo Ministry of the Environment	Air quality by sensor in Tokyo	location
Tokyo Parks	Tokyo Bureau of Construction	Park statistics by Ku	Ku
Tokyo Crime Data	Tokyo Metropolitan Police	Crime statistics by administrative district in Tokyo (chome)	location
Land Use by District	Tokyo Bureau of Urban Development	Land use classification breakdown in Tokyo by Ku	Ku
Complaints about Pollution by Kind	Tokyo Ministry of the Environment	Total complaints about pollution by Ku	Ku

4.3.2 Data Pre-Processing - Geospatial and Environmental Model

The geospatial and environmental model uses the baseline model's features with the addition of features from the environmental datasets' listed in section 2. Table 12 details the datasets used in this model as well as descriptions and common keys the data were merged on. One major change to this model was how travel time to stations was encoded. A new feature, Time_to_station was created to represent the aggregate travel time to all stations by walking, bus, and car. This was done because some rental properties actually require the usage of a combination of travel methods. New binary features WalkFlag, BusFlag, and CarFlag were created for each station to indicate whether a rental property required that mode of transport to a station.

4.3.3 Data Pre-Processing - Feature Engineering Model

The feature engineering model takes all of the features from the first two models and adds additional new features based on intuition and experience to boost performance. Correlation is often a measure used to represent relationship between a feature and the target. It will be calculated against the training set target as it can help evaluate whether an engineered feature is suitable for a model. The types of feature engineering techniques used to generate features in this study can be broken into the following categories which will be discussed further in detail.

- Binary Encoding
- Expansion Encoding
- Consolidation Encoding
- Interaction Features
- Ranking Features
- Aggregate Features
- Target-Mean Encoding
- Geospatial Coordinate System Projection

Binary Encoding

Binary Encoding is a method for representing features based on a conditional. Data points that meet a condition are encoded as 1's while others are represented as 0's. One binary feature was created in this study titled first_floor_FE which encodes all first floor listings as its own feature. It is commonly known that living on the first floor in Tokyo is typically cheaper due to safety and other concerns.

Expansion Encoding

Expansion encoding is a technique for creating multiple features from one feature based on the pieces of information contained in the data. Room_layout is a feature that contains layouts like 1LDK (living room, dining room, kitchen) or 1K (kitchen) which indicate multiple pieces of information including the number of rooms and types of rooms available. This feature was expanded to create 6 separate features representing the number of rooms as well as whether the property had a living room, dining room, kitchen, service room, or was just a 1 room layout. We can see based on correlation that having a separate living room, dining room, and multiple rooms correlated highly with the target value while living in only one room was negatively correlated. New features generated by expansion encoding are detailed in table 13. Although this method is called expansion encoding, this method reduces the final dataset dimensionality as each unique room layout is no longer one-hot-encoded.

Consolidation Encoding

Consolidation encoding is a feature engineering method that maps multiple categorical variables to the same variable. The 23 Ku's in Tokyo are already large areas but these areas can be consolidated further into groups depending on

Table 13: Expansion Encoding Features (room_layout)

New Feature	Description	Correlation to Target
total_rooms_FE	Total number of rooms	0.368
Living_FE	Living room?	0.595
Dining_FE	Dining room?	0.469
Kitchen_FE	Separate kitchen?	0.218
Service_Room_FE	Service room?	0.194
Room_Only_FE	1 room layout	-0.218

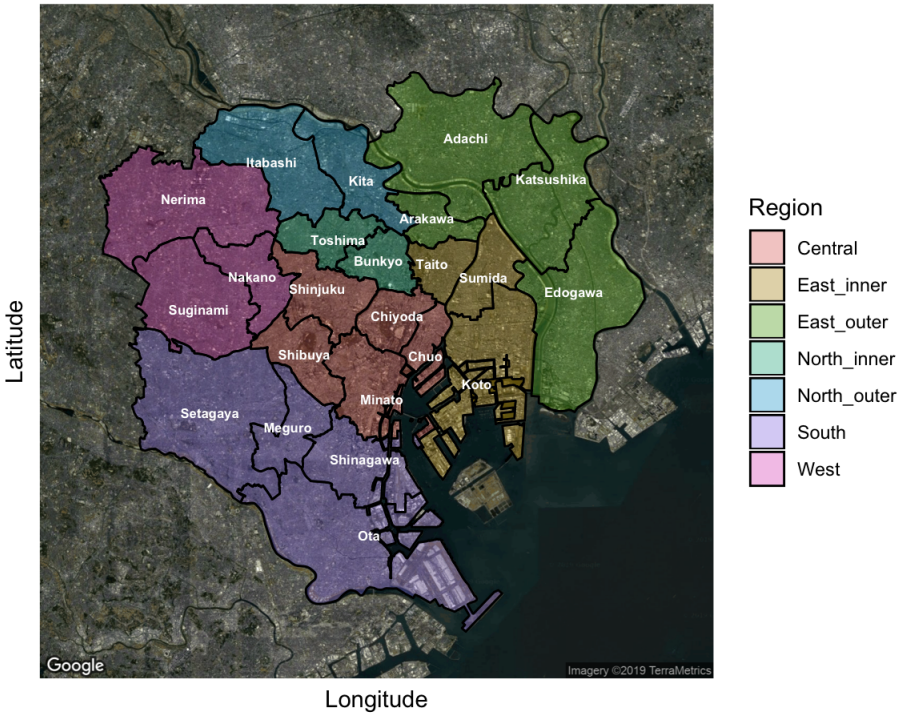


Figure 7: Consolidation Encoding Map

their geospatial region of Tokyo. It is hypothesized that different regions of Tokyo value different hub stations so it is important to represent this as a feature in some way. The 23 Ku's are consolidated and mapped to the regions North_outer, North_inner, West, East_other, East_inner, Central, and South depending on geographic location. The mapping for each Ku for the new Region_FE feature can be seen in figure 7 .

Interaction Features

Interaction features are created by performing mathematical operations on two or more features. Interaction features allow us to capture or encode the unknown interaction that may exist between features. This technique is especially effective for encoding the interactions between features from different datasets or features with high correlation to the target. Features in this section were constructed based on the strongest performing features and correlations of the second geospatial environmental model. This study creates a large number of interaction features which are detailed further in tables 14, 15, 16, 17, 18, 19, 20:

Rank Encoding

Rank Encoding features are created by calculating the rank or position of a data point in the overall dataset. Crime was ranked for each administrative district to get the overall rank of crime in the dataset. This type of feature can tell us how dangerous a specific location is in comparison to other areas. The rankings were calculated on the main categories of crime including total crime, felony's, violent crimes, burglary and larceny, non-intrusive larcenies, and other crimes. It doesn't seem these features will have much impact based on correlation but it's possible these rankings will provide

Table 14: Interaction Features (Building)

Feature Name	Description	Correlation to Target
area_per_room_FE	area / total_rooms	-0.056
height_ratio_FE	floor / Height	-0.086
area_height_FE	area × Height	0.549
floor_area_FE	floor × area	0.499

Table 15: Interaction Features (Risk)

Feature	Description	Correlation to Target
yrs_risk_FE	yrs × Total_Risk	-0.180
yrs_lifediff_FE	yrs × Life_Difficulty_Risk	-0.225
yrs_fire_FE	yrs × Fire_Risk	-0.153
yrs_building_FE	yrs × Building_Risk	-0.248

Table 16: Interaction Features (Transit)

Feature	Description	Correlation to Target
Hub_Aggregate_FE	Sum of transit times to major hubs (made features)	-0.313
Shinjuku_Commute_FE	Transit time to Shinjuku + Time to Station_1	-0.213
Tokyo_Commute_FE	Transit time to Tokyo + Time to Station_1	-0.301
Shinagawa_Commute_FE	Transit time to Shinagawa + Time to Station_1	-0.306
Ueno_Commute_FE	Transit time to Ueno + Time to Station_1	-0.254
Ikebukuro_Commute_FE	Transit time to Ikebukuro + Time to Station_1	-0.072
Shibuya_Commute_FE	Transit time to Shibuya + Time to Station_1	-0.356
Aggregate_Commute_FE	Sum of commutes to all major hubs	-0.297

Table 17: Interaction Features (Interaction x Interaction)

Feature	Description	Correlation to Target
agg_commute_floor_area_FE	Aggregate_Commute_FE × floor_area_FE	0.422
agg_commute_area_height_FE	Aggregate_Commute_FE × area_height_FE	0.485
agg_commute_area_per_room_FE	Aggregate_Commute_FE × area_per_room_FE	-0.290
agg_commute_yrs_building_FE	Aggregate_Commute_FE × yrs_building_FE	-0.267
agg_commute_yrs_fire_FE	Aggregate_Commute_FE × yrs_fire_FE	-0.156
agg_commute_yrs_lifediff_FE	Aggregate_Commute_FE × yrs_lifediff_FE	-0.253
agg_commute_yrs_risk_FE	Aggregate_Commute_FE × yrs_risk_FE	-0.183

Table 18: Interaction Features (Land Classification)

Feature	Description	Correlation to Target
Residential_land_ratio_FE	Residential / Total_District_Land	0.050
OtherUse_land_ratio_FE	OtherUse / Total_District_Land	-0.287
Parks_land_ratio_FE	Parks / Total_District_Land	-0.016
Fields_land_ratio_FE	Fields / Total_District_Land	-0.231
WoodForest_land_ratio_FE	WoodsForest / Total_District_Land	-0.063
Water_land_ratio_FE	Water / Total_District_Land	-0.021
Farm_land_ratio_FE	Farm / Total_District_Land	-0.191
Roads_land_ratio_FE	Roads / Total_District_Land	0.110
Unused_land_ratio_FE	Unused / Total_District_Land	0.157

Table 19: Interaction Features (Pollution Complaints)

Feature	Description	Correlation to Target
Air_Pollution_Complaints_per_person_FE	Air_Pollution / Population	-0.137
Water_Pollution_Complaints_per_person_FE	Water_Pollution / Population	0.237
Soil_Pollution_Complaints_per_person_FE	Soil_Pollution / Population	-0.067
Other_Pollution_complaints_per_person_FE	Other_Pollution / Population	0.120
Offensive_odors_Complaints_per_person_FE	Offensive_odors / Population	0.136
Vibration_Complaints_per_person_FE	Vibration / Population	0.070
Noise_Pollution_Complaints_per_person_FE	Noise / Population	0.247

Table 20: Interaction Features (Crime)

Feature	Description	Correlation to Target
Felony_Offense_Ratio_FE	Felonious_Offense_Total / Total_Crimes	0.026
Violent_Crime_Ratio_FE	Violent_Crime_Total / Total_Crimes	0.111
Burglary_Larceny_Ratio_FE	Burglary_Larceny_Total / Total_Crimes	0.009
Non_Intrusive_Larceny_Ratio_FE	Non_Intrusive_Larceny / Total_Crimes	-0.116
Other_Crime_Ratio_FE	Others_Total / Total_Crimes	0.081

more context into each administrative district than the simple raw numbers provided with the data. Features generated through rank encoding are listed in table 21.

Categorical Binning

Categorical binning is a method for grouping of categories based on their characteristics. In this research, the stations and train lines without many samples are grouped together to make a minority category. In addition, this type of binning is performed to bin administrative districts' in each Ku with not many rental listings resulting in 23 distinct minority categories. These categories with not so many samples often negatively impact models which grouping can help with. These new minority categories created within existing features as well as their encoding thresholds can be seen in table 22.

Target-Mean Encoding

Target mean encoding is a feature encoding method that codes a categorical value by its expected value of the target. Only samples in the training set are used and due to its use of the target value, it does have some risk of overfitting. The most basic way to perform target encoding is to calculate the mean for every category in a feature and replace the category with that mean based on the training set. This research employs a form of target encoding that uses the prior probability as a regularization method [37]. The prior probability and categorical probability are blended with the help of two hyperparameters, smoothing and min_samples_leaf. These parameters control the balance between categorical average and prior as well as the minimum required samples used to take the category average into account during blending. The aim of this method is to weigh the categorical average more in a category with higher samples and to weigh more towards the prior in categories with less samples. After this mean based on these two hyperparameters is calculated, a little noise is added and used to impute each category in the training, validation, and test set.

One of target encodings' key benefits is that it can reduce a categorical variable to 1-dimension in contrast to one-hot-encoding. In this research, all categories with high cardinality were encoded with target mean encoding including the

Table 21: Rank Encoding (Crime)

New Feature	Description	Correlation to Target
Crime_Rank_FE	Ranking by location based on total overall crime	-0.029
Felony_Rank_FE	Ranking by location based on total felonies	-0.057
Violent_Crime_Rank_FE	Ranking by location based on total violent crimes	-0.082
Burglary_Larceny_Rank_FE	Ranking by location based on total burglary and larceny	0.009
Non_Intrusive_Larceny_Rank_FE	Ranking by location based on total non intrusive larceny	-0.003
Other_Crime_Rank_FE	Ranking by location based on other crimes	-0.063

Table 22: Categorical Binning

Feature	Category Mapping	Threshold
location	(Ku)_Minority (23 Categories)	< 5 samples
Line_1	Line_Minority	< 30 samples
Station_1	Station_Minority	< 30 samples
Line_2	Line_Minority	< 30 samples
Station_2	Station_Minority	< 30 samples
Line_3	Line_Minority	< 30 samples
Station_3	Station_Minority	< 30 samples

Table 23: Coordinate System Projections

Features	Source LAT/LON
x_cart, y_cart, z_cart	location
x_cart_Eki1, y_cart_Eki1, z_cart_Eki1	Station_1
x_cart_Eki2, y_cart_Eki2, z_cart_Eki2	Station_2
x_cart_Eki3, y_cart_Eki3, z_cart_Eki3	Station_3

features location, Line_1, Line_2, Line_3, Station_1, Station_2, and Station_3. This research sets a smoothing value of 10, noise of .001, and min_samples_leaf are selected based on the total number of samples at the 20th percentile category for each feature based on experimentation.

Spatial Coordinate System Projection

Although latitude and longitude data exists for all locations and stations, coordinate system projections are an interesting way to augment spatial data representations. This technique was experimented with in this research because many data science competitions involved with geospatial data have used coordinate system projection techniques with good results. Converting latitude and longitude to Cartesian coordinates allows us to convert a 2-dimensional representation of data to a 3-dimensional space. Four sets of x, y, z Cartesian coordinates were created based on the coordinates of the administrative district (location) and 3 stations which is detailed in table 23.

4.4 Final Pre-Processing Steps

For each model, data type downcasting was performed to reduce the memory usage of each feature to promote quicker model training and experiments. In addition, feature scaling and category encoding were performed on numeric and categorical features as described below.

Feature Scaling

The Scikit-Learn Standard Scaler was used to scale all numeric features. Standardization is a common processing step for machine learning algorithms before modeling and is used to re-scale all features to the same scale. Standardization gives each feature a mean and unit variance of 0 which more closely mimics the properties of a Gaussian distribution allowing for each feature to be equally represented in terms of scale during the learning process.

Category Encoding

All category's were one hot encoded to create individual features for each distinct category with the exception of the feature engineering model which uses a mixture of target encoding and one hot encoding depending on cardinality of the categorical features. One hot encoding is the most basic method for handling of categorical variables and represents each unique category in a feature as its own binary representation. This has one drawback of increasing the amount of features by a factor of the total unique categories potentially putting increased demands on computation. This transformation results in 5,277 additional features for the baseline model and 5,287 features for the geospatial and environmental model when factoring in all the unique values for location, floor_plan, Ku, Line_1, Line_2, Line_3, Station_1, Station_2, Station_3, and Land Classification. Since categorical features with high cardinality use the target encoding method in the feature engineering model, only 56 new features are created via one hot encoding. This is a significant reduction in features through target mean encoding.

Final Breakdown

Table 24: Feature Breakdown for Each Model

Model	Categorical Feature Encoding Method	Numeric Columns	Encoded Categorical Features	Total Features
Baseline (1)	One-Hot-Encoding	22	5,277	5,299
Geospatial and Environmental (2)	One-Hot-Encoding	207	5,287	5,494
Feature Engineering (3)	One-Hot-Encoding/Target Mean Encoding	283	56	339

After performing feature scaling and one hot encoding, the data is split into 6 sparse matrices for each experiment labeled as X_train, y_train, X_valid, y_valid, X_test, and y_test. This was done to optimize computation speed in the training and hyperparameter optimization steps. Sparse matrices have many zero values which allow for compression of memory and speedup of algorithms. The baseline model after transformation contains 22 numeric features and 5,277 transformed categorical features resulting in a total of 5,299 features. The geospatial and environmental model contains 207 numeric values and 5,287 transformed categorical features for a total of 5,494 features. The final feature engineering model contains 283 numeric features and 56 transformed categorical features resulting in a total of 339 features. Table 24 details the complete feature breakdown by model.

4.5 LightGBM/Gradient Boosting

The algorithm chosen for these experiments is LightGBM which is an algorithm from the Gradient Boosting Decision Trees (GBDT) family developed by Microsoft [38]. A Gradient Boosting Decision Tree (GBDT) is an ensemble model that uses multiple weak learners which are trained sequentially to make a stronger model. This comes from a mathematical technique called additive modeling where a composite function is created through the addition of simpler sub-functions. In the case of GBDT, these sub-functions are individual decision trees where summed up are represented as:

$$y(\hat{x})^K = \sum_{i=1}^{K} f_i(x) \tag{6}$$

where f_i is the output of the ith of regression tree of the Kth ensemble and $y(\hat{x})^K$ is the predicted output. Gradient Boosting methods typically train using the residual errors to improve sequentially at every step. To build the $(K + 1th)$ tree a regularized objective function is minimized defined as:

$$L = \sum_{i=1}^{n} L(y_i, \hat{y}_i^K + f_{K+1}(x_i)) + \Theta(f_{K+1}) \tag{7}$$

where $L(y_i, \hat{y}_i^K + f_{K+1})$ represents the calculated mean squared error between the actual target and the predicted target in the next tree. Θ represents a regularization term that penalizes complexity and controls overfitting.

LightGBM has become popular recently due to its success in many machine learning competitions including the Recruit Visitor Forecasting and Mercari Price Prediction among others [40]. Recent experiments also show it as generally being able to converge to a solution that generalizes better which is suited to a large and highly multidimensional dataset like this [41] .

The other state of the art GBDT's include XGBoost and CatBoost. Although these algorithms are implementing similar functionality recently, there are a few key differences that existed when they were first released. One of the key challenges with optimization of decision trees is determining how to find the optimal splits. In the case of XGBoost, pre-sort-based algorithms are used for decision tree learning while LightGBM's contribution relied on histogram-based algorithms to bucket continuous features values into discrete bins. LightGBM also uses a leaf wise growth strategy in contrast to level (depth) wise growth that XGBoost utilizes by default. This method for calculating splits is said to achieve lower loss than depth wise growth. Figure 8 illustrates level-wise vs leaf-wise growth. Since an objective function and evaluation function are defined, early stopping of training will occur whenever either the MSE or RMSLE metric does not improve for 1000 tree iterations.

4.6 Hyperparameter Optimization

Hyperparameter optimization is the process of finding the optimal hyperparameters for a learning algorithm. Although LGBM is quick and generally accurate, one challenge with its use is the huge number of hyperparameters the algorithm

(a) Level Wise Tree Growth (b) Leaf Wise Tree Growth

Figure 8: Level vs Leaf Wise Tree Growth [39]

Table 25: Bayesian Optimization Hyperparameter Search Range and Final Parameters (rounded to 2 decimals)

Parameter	Search Range	Baseline	Geo_Env	FE
bagging_fraction	0.8-1	0.99	0.95	0.98
colsample_by_tree	0.5-0.7	0.59	0.66	0.68
feature_fraction	0.1-0.9	0.76	0.57	0.86
lambda_l1	0-5	4.31	2.23	0.31
lambda_l2	0-3	2.07	1.21	2.63
learning_rate	.05-0.2	0.16	0.17	0.13
max_depth	5-9	8	9	7
min_child_weight	5-50	29.69	47.42	17.67
min_split_gain	0.001-0.1	0.03	0.06	0.01
num_leaves	24-45	25	40	41
subsample	.5-.7	0.59	0.52	0.63
device_type	gpu	gpu	gpu	gpu
early_stopping_rounds	1000	1000	1000	1000

has. The optimal hyperparameters for the LightGBM models were chosen through use of bayesian optimization and can be found for each respective model in table 25 [42].

Bayesian optimization is a method for hyperparameter optimization that uses probabilistic Gaussian processes to find the optimal hyperparameters. Bayesian optimization is generally guided by Gaussian processes to approximate the objective function as well as an acquisition function that selects the next hyperparameter point to evaluate based on information from previous samples. This allows bayesian optimization to find the optimal parameters quicker than most hyperparameter optimization methods. The acquisition function used in this optimization was the expected improvement function which is considered standard in terms of performance. Bayesian optimization was run with 5-fold cross-validation on the training set and optimal parameters were tested and selected based on the highest performing hyperparameters' scores on the validation set. Bayesian optimization was executed for each model until a plateau was reached where cross-validation scores no longer improved.

5 Computational Results

This research quantifies the effectiveness of the use of feature engineering and environmental features by comparing the baseline model to the two constructed models. Overall model performance is reported in table 26 and 27 which details the results of 5-fold cross validation (5x) and test set evaluation for the MAPE and RMSE metrics.

Although the baseline model performed well as a benchmark and contributed to a majority of the predictive performance, each consecutive model was able to exceed the baseline performance through the use of additional features.

The biggest increase in performance came with the addition of geospatial and environmental features. The second model was able to improve the MAPE score from 5.91 % (baseline) to 5.32 % (geo-env) and RMSE score from 12,494 yen (baseline) to 11,692 yen (geo-env) which is a significant reduction in error. This shows the effectiveness supplemental environmental features can have even with minimal pre-processing in the context of rent prediction in Tokyo.

The proposed feature engineering approach greatly improves all proposed evaluation metrics and reduces the total number of features required for model training. Compared to baseline, the MAPE score was reduced from 5.91 % (baseline) to 5.01 % (FE) which even improves upon the geospatial and environmental model. The RMSE also improved

Table 26: MAPE Scores by Model

Evaluation Method	Baseline	Geo-Env	Feature Engineering
5-Fold CV (x5)	5.89	5.36	5.12
Test Set Evaluation	5.91	5.32	5.01

Table 27: RMSE Scores by Model

Evaluation Method	Baseline	Geo-Env	Feature Engineering
5-Fold CV (x5)	12,494.14	11,692.04	11,269.30
Test Set Evaluation	12,204.35	11,328.02	11,202.46

in comparison to the baseline model but the change was not so drastic in comparison to the geospatial and environmental model's impact. The combination of feature engineering with all features yielded the highest performance for all metrics and shows that using using domain knowledge to model and extract complex relationships between features should not be neglected as these little changes can give the slight boost in performance needed to create a model that can generalize better.

5.1 Ku Comparison

The Geo-Env model and FE model both reduce the MAPE scores significantly for each Ku in Tokyo showing their effectiveness in each part of Tokyo. The most significant changes or decreases of MAPE score from the Geo-Env model were seen in Arakawa, Katsuhika, Nerima, Setagaya, Kita, and Suginami-Ku. The feature engineering model lead to the most positive changes in Arakawa, Katsushika, Nakano, Meguro, and Suginami-Ku. A detailed breakdown of each Ku's MAPE scores by model and reductions can be found in table 28.

When analyzing the RMSE score for each Ku, it is evident that some areas are harder to predict than others. Minato, Chiyoda, Chuo, and Shibuya-Ku located in central tokyo are difficult to predict with higher average target values and are known to have more luxury properties. However, when looking at the MAPE score we can see that predictions in these areas are still within around 5 % despite the presence of luxury and expensive properties. It was interesting to see that some areas like Bunkyo, Minato, and Chiyoda-Ku suffered from higher RMSE scores which could indicate that new features added some noise negatively impacting prediction in these areas. It is possible that areas like Minato and Chiyoda- Ku are not so influenced by environmental factors as many properties in these areas are considered to be out of the norm in comparison to most of Tokyo. For the majority of Tokyo's 23 special wards, geospatial, and environmental data along with feature engineering was able to decrease the RMSE contributing to more effective rent estimation as seen in table 29.

5.2 Price Bin Comparison

Also evaluated in this study were the 15 price groupings used to perform stratified splitting on the datasets. As evidenced by table 30, the second model was able to reduce the average prediction error in every single price grouping. The feature engineering model also improved on every single price segment in comparison to the Geo-Env model. All models did not perform very well on the lowest pricing segments possibly due to the presence of outliers. It was interesting to see that the biggest reduction in MAPE scores was in the lowest price bin groups. These results indicate that geospatial and environmental features can be very effective for improvement of predictions within all price groups but especially in lower price brackets.

RMSE scores also followed a similar trend with significant reductions through the addition of features and further reduction through feature engineering as observed in table 31. Although outperforming the baseline for nearly every price segment, the feature engineering model did not perform as well as the second model on the most expensive rent price segment.

5.3 Feature Importance

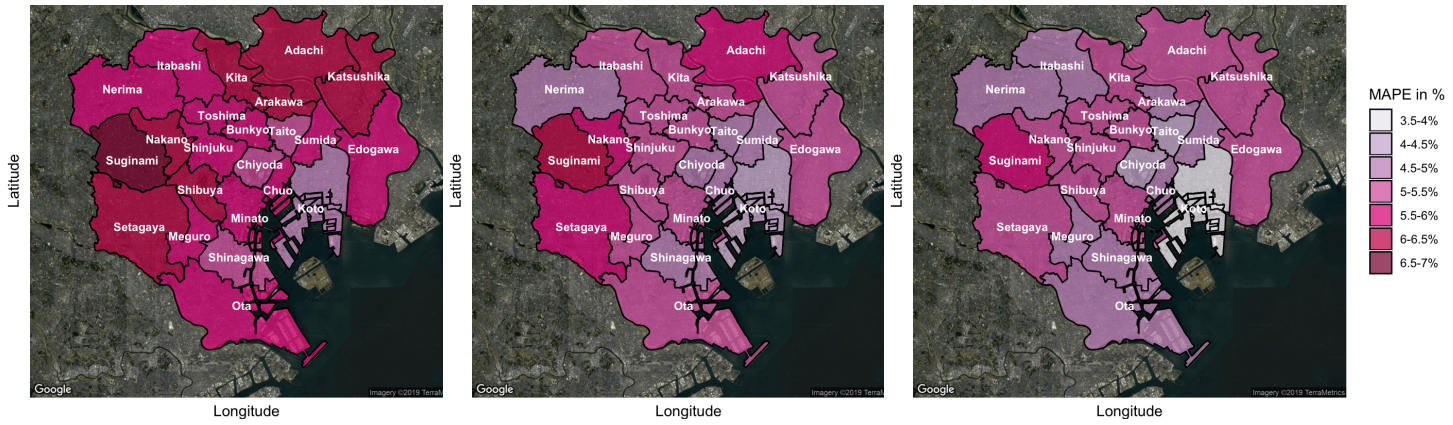
Interpretive machine learning has become very important recently which is the ability to understand why and how a model makes a prediction. This research attempts to make sense of why the models have improved in comparison to the baseline with mechanisms provided by LightGBM for feature interpretation.

Table 28: MAPE by Model and Comparison to Baseline by Ku

Ku	Baseline	Geo-Env	FE	Geo-Env Change	FE Change
Arakawa	6.28	5.38	5.03	-0.90	-1.25
Katsushika	6.22	5.33	5.12	-0.89	-1.10
Nakano	6.44	5.89	5.34	-0.55	-1.10
Meguro	5.63	5.07	4.57	-0.56	-1.06
Suginami	7.00	6.35	5.96	-0.65	-1.04
Setagaya	6.29	5.56	5.25	-0.73	-1.03
Kita	6.07	5.37	5.06	-0.70	-1.01
Nerima	5.68	4.84	4.67	-0.84	-1.00
Adachi	6.09	5.57	5.14	-0.52	-0.96
Ota	5.89	5.30	5.00	-0.59	-0.89
Chiyoda	5.09	4.52	4.22	-0.57	-0.87
Shibuya	6.13	5.49	5.27	-0.64	-0.86
Sumida	5.53	4.95	4.69	-0.58	-0.83
Koto	4.75	4.22	3.94	-0.53	-0.81
Chuo	5.54	4.91	4.73	-0.63	-0.80
Shinagawa	5.40	4.83	4.59	-0.56	-0.80
Edogawa	5.90	5.35	5.13	-0.54	-0.77
Taito	5.20	4.68	4.45	-0.51	-0.75
Itabashi	5.70	5.12	4.99	-0.59	-0.71
Toshima	5.79	5.40	5.11	-0.39	-0.68
Minato	5.69	5.21	5.06	-0.49	-0.63
Bunkyo	5.97	5.47	5.37	-0.49	-0.59
Shinjuku	5.71	5.44	5.12	-0.27	-0.59

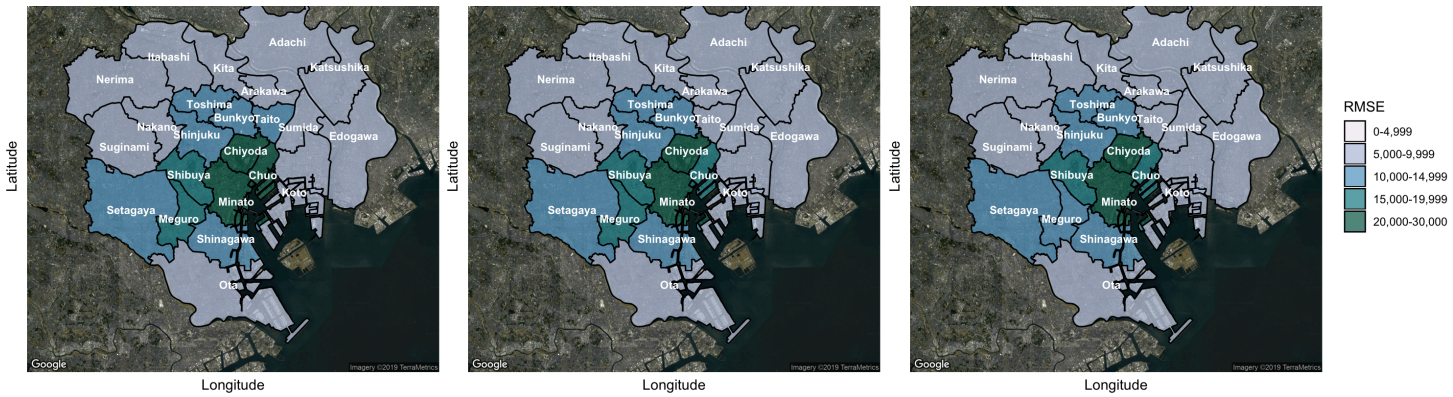
Table 29: RMSE by Model and Comparison to Baseline by Ku

Ku	Baseline	Geo-Env	FE	Geo-Env Change	FE Change
Meguro	16,108	15,437	13,272	-671	-2,836
Chuo	22,438	19,728	19,804	-2,710	-2,634
Shibuya	19,083	15,849	16,520	-3,235	-2,563
Taito	11,922	9,915	9,512	-2,007	-2,410
Toshima	12,630	10,853	10,384	-1,777	-2,246
Setagaya	12,054	10,623	10,074	-1,431	-1,980
Nerima	9,134	7,814	7,292	-1,320	-1,842
Arakawa	9,387	8,079	7,923	-1,308	-1,464
Kita	8,961	8,300	7,654	-661	-1,307
Ota	9,440	8,483	8,199	-957	-1,240
Edogawa	8,147	7,527	6,966	-620	-1,182
Shinagawa	13,447	11,027	12,404	-2,420	-1,043
Sumida	9,146	8,531	8,157	-614	-989
Itabashi	7,745	6,847	6,908	-899	-837
Suginami	9,653	9,202	8,841	-450	-812
Katsushika	7,135	6,761	6,382	-374	-753
Koto	9,328	8,932	8,581	-395	-747
Nakano	8,744	8,697	8,017	-47	-727
Shinjuku	13,654	13,304	13,234	-350	-421
Adachi	7,448	7,983	7,087	535	-361
Bunkyo	12,114	12,193	12,863	79	749
Minato	23,681	24,291	24,456	610	775
Chiyoda	21,584	21,842	25,742	258	4,158



(a) Baseline (b) Geo-Env Model (c) FE Model

Figure 9: MAPE scores for each Ku by Model



(a) Baseline (b) Geo-Env Model (c) FE Model

Figure 10: RMSE scores for each Ku by Model

LightGBM has a function for calculating feature importance by total gain and total splits. The total gain measures the relative contribution of a feature in comparison to others. This measure can often be an indicator for how much a feature contributes to a prediction. Total splits on the other hand measures how many times a feature is used in the decision tree splitting criteria. Although it does have some contribution to tree construction it does not always translate to predictive power. This section will analyze feature importance through the gain and split metrics observed in all models.

It is interesting to note that all models had high gain and split scores for the area, yrs, floor, and height features. This indicates how fundamental these standard rental descriptive features are to rent prediction. Area has a high influence on rent in Tokyo and it is well known that many people pay a premium for more space which explains why it takes the top gain in every model. The age of an apartment is an additional indicator for disaster risk in Tokyo as building codes change and price tends to depreciate rapidly with age. Tower mansions also typically have many more floors than traditional apartments which come at a price premium for residents.

For the geospatial and environmental model, features like time and distance to Shibuya, park related features, as well as building risk entered the top 15 list for gain. When analyzing the feature engineering model, 11/15 of the features (mainly interaction features) with the highest gain were created manually which shows the effectiveness feature engineering played in the performance of the last model. Target encoding of the location ranked second for gain which is logical from the standpoint of rent modeling as it is a feature representative of the average price of an administrative district. Other interesting engineered features with high gain include whether an apartment has a living room, the interaction between area and height, as well as the total area per room.

The most interesting and surprising feature to have a high gain score was the WS(ms)_std which is a measure for wind speed standard deviation. Although wind speed does have an impact on air quality and pollution, further analysis reveals

Table 30: MAPE by Model and Comparison to Baseline by Price Group

Price Group	Baseline	Geo-Env	FE	Geo-Env Change	FE Change
23,999-53,000	10.13	9.2	8.94	-0.93	-1.19
53,000-60,000	6.78	6.13	5.56	-0.65	-1.22
60,000-66,000	6.27	5.67	5.12	-0.6	-1.15
66,000-72,000	5.59	5.16	4.8	-0.43	-0.79
72,000-77,000	5.81	5.29	5	-0.52	-0.81
77,000-82,500	5.67	5.18	4.69	-0.49	-0.98
82,500-88,000	5.52	5.03	4.88	-0.49	-0.64
88,000-95,000	5.72	5.12	4.88	-0.59	-0.83
95,000-102,000	5.2	4.6	4.21	-0.6	-0.99
102,000-110,000	5.09	4.61	4.29	-0.48	-0.81
110,000-121,000	4.69	4.07	3.89	-0.62	-0.8
121,000-135,000	4.85	4.33	4.06	-0.52	-0.79
135,000-155,000	5.27	4.61	4.36	-0.66	-0.91
155,000-203,000	5.7	4.97	4.74	-0.73	-0.95
203,000-790,000	6.36	5.63	5.63	-0.73	-0.73

Table 31: RMSE by Model and Comparison to Baseline by Price Group

Price Group	Baseline	Geo-Env	FE	Geo-Env Change	FE Change
23,999-53,000	6,412	6,103	5,824	-309	-588
53,000-60,000	5,369	5,067	4,670	-302	-699
60,000-66,000	5,469	5,201	4,683	-267	-785
66,000-72,000	5,606	5,533	4,998	-73	-609
72-000-77,000	6,323	5,893	5,562	-430	-761
77,000-82,500	6,430	6,267	5,575	-163	-855
82,500-88,000	6,920	6,664	6,442	-256	-478
88,000-95,000	8,090	7,852	7,252	-237	-838
95,000-102,000	7,609	7,155	6,490	-455	-1,119
102,000-110,000	7,865	7,515	7,025	-350	-840
110,000-121,000	8,376	7,524	7,288	-852	-1,088
121,000-135,000	9,972	8,987	8,416	-986	-1,557
135,000-155,000	11,939	10,912	10,649	-1,027	-1,290
155,000-203,000	16,523	15,495	14,564	-1,027	-1,959
203,000-790,000	34,862	32,090	33,051	-2,772	-1,811

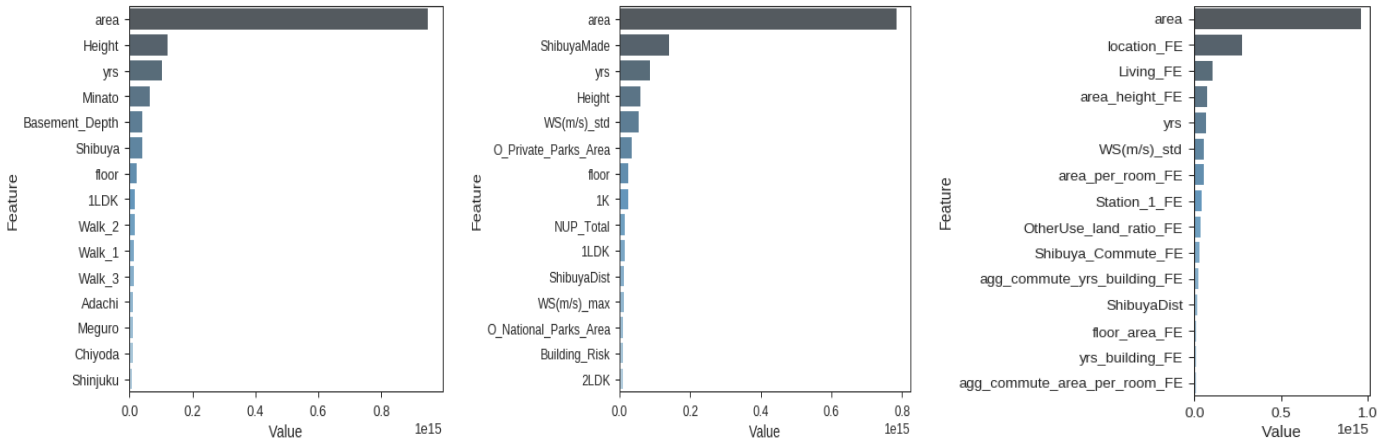
that wind speeds generally had a higher standard deviation in lower priced Ku's and lower standard deviation around the center of Tokyo. This suggests that this feature functions as another proxy for location in Tokyo which could be an explanation for the features importance in model construction.

When approaching this from a total splits perspective we can see many splits in the baseline were in features like area, yrs, floor, Height, and walking time to stations. For the Geo-Env model the same features had high splits with the addition of geospatial attributes like longitude and latitude as well as disaster risk and crime features. Around 13/15 of the top 15 features were manually created in the feature engineering model through target encoding and interaction features that combined building, environmental, and geospatial attributes.

6 Future Work and Conclusion

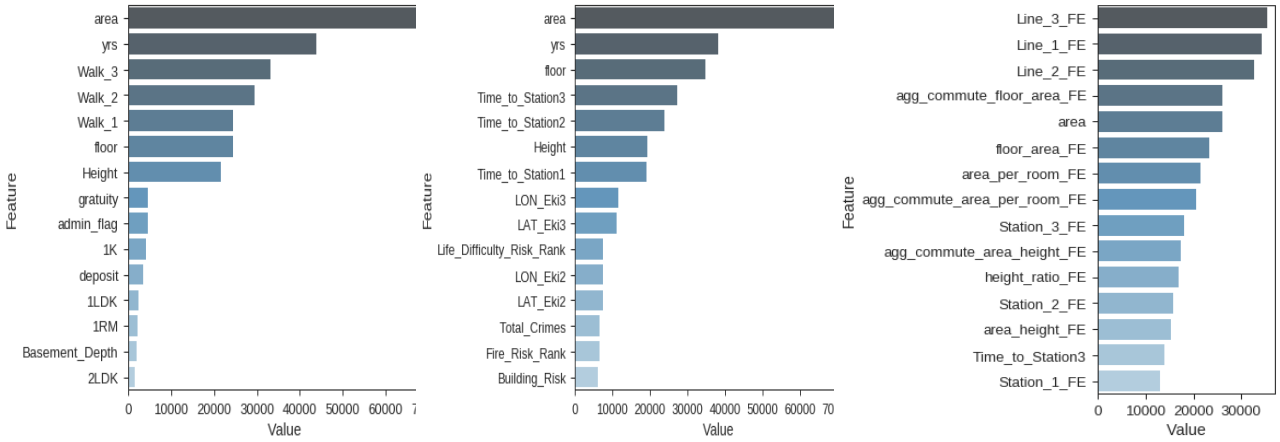
6.1 Future Work

This research builds a model for rent estimation in Tokyo but could also likely be applied to housing price and land estimation around the world. Possible future works include deploying a similar rent estimation model in other regions in Japan as market dynamics differ vastly between countryside areas and Tokyo. Tokyo can be considered to be an outlier and special case as nowhere else in Japan has a real estate market as challenging. In addition, ensemble modeling may be able to push the model's accuracy even further as evidenced by other research. Feature selection was not employed



(a) Baseline (b) Geo-Env Model (c) FE Model

Figure 11: Feature Importance by Gain for Each Model



(a) Baseline (b) Geo-Env Model (c) FE Model

Figure 12: Feature Importance by Total Splits for Each Model

in this study which also serves as an opportunity for model improvement. Building separate models for different market segments or further removal of outliers may also be an interesting avenue to pursue as the biggest challenge in this research was prediction of the lower and upper quantiles.

In addition, it would be interesting to utilize neural networks on existing unstructured data like text, room layout images, and building external images to perform further feature extraction. Key features like amenities in neighborhoods and extras like building structure material, air conditioning, and security features among many others would definitely improve performance if available. In regards to categorical cardinality, target mean encoding was used for this study which often suffers from overfitting as it incorporates the target value. Another recently proposed method called categorical entity embedding may be useful to try and overcome the categorical dimensionality gap and overfitting issue.

Real estate modeling can use an infinite number of features to describe an area so any data that is available can be beneficial to overall modeling if pre-processed, engineered, and applied smartly. The local environment is only one data source but there are surely many more undiscovered sources as society continues to collect and process more data than ever before.

6.2 Conclusion

This paper presented a time independent gradient boosting approach to predicting rent in Tokyo through the use of geospatial and environmental features as well as feature engineering techniques. In real estate market estimation, it is possible that even the smallest percentage in performance improvement can be quantified to have a huge impact on a businesses profit. This study provides valuable insights into the viability of environmental features as well as the effectiveness of feature engineering techniques when applied to a common real estate modeling problem like rent prediction. Feature engineering is often associated with the generation of more features but when applied in a crafty way can be used to reduce features and maintain accurate representation of the data with minimal loss of information. Real estate price prediction is a difficult problem due to the fact that an infinite number of features could be used to augment a model and improve performance. This research hopes that more researchers will look to the natural environment of an area and apply feature engineering techniques when building models for real estate estimation as well as other domains.

Acknowledgements This research would not be possible without the generous support of The Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT). I also appreciate my advisor Yuichiro Miyamoto who was patient and helpful despite my lack of Machine Learning experience when entering his laboratory.

References

[1] Housing. OECD Better Life Index. http://www.oecdbetterlifeindex.org/topics/housing/.

[2] 68% of the world population projected to live in urban areas by 2050, says UN. (2018, May 16). United Nations. UN DESA Department of Economic and Social Affairs. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html

[3] Zillow, Inc. Real Estate, Apartments, Mortgages Home Values. Zillow. https://www.zillow.com/

[4] Property valuations engineered for the modern world. GeoPhy. https://geophy.com/.

[5] Make smarter real estate decisions. PriceHubble. https://www.pricehubble.com/en/

[6] SUUMO. online platform in Japan for housing, real estate buying and selling, and rental support information. https://suumo.jp/[Japanese]

[7] Zillow, Inc. What is a Zestimate? Zillow's Zestimate Accuracy. Zillow. https://www.zillow.com/zestimate/

[8] Zillow, Inc. Zillow Prize Winners. Zillow. https://www.zillow.com/marketing/zillow-prize/

[9] Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. Journal of Political Economy. https://doi.org/10.1086/260169

[10] Mullainathan, S., Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. Journal of Economic Perspectives. https://doi.org/10.1257/jep.31.2.87

[11] Jiang, L.; Phillips, P.C.B.; Yu, J. A New Hedonic Regression for Real Estate Prices Applied to the Singapore Residential Market. Technical Report, Cowles Foundation Discussion Paper No. 1969, 2014.

[12] SELİM, S. (2019). Determinants of House Prices in Turkey : A Hedonic Regression Model. Doğuş Üniversitesi Dergisi. https://doi.org/10.31671/dogus.2019.223

[13] Owusu-Ansah, A. (2013). A review of hedonic pricing models in housing research. A Compendium of International Real Estate and Construction Issues.

[14] Oladunni, T., Sharma, S. (2017). Hedonic housing theory - A machine learning investigation. Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016. https://doi.org/10.1109/ICMLA.2016.103

[15] Mayer, Michael and Bourassa, Steven C. and Hoesli, Martin Edward Ralph and Scognamiglio, Donato Flavio, Estimation and Updating Methods for Hedonic Valuation (December 12, 2018). Swiss Finance Institute Research Paper No. 18-76. Available at SSRN: https://ssrn.com/abstract=3300193 or http://dx.doi.org/10.2139/ssrn.3300193

[16] Baldominos, A.; Blanco, I.; Moreno, A.J.; Iturrarte, R.; Bernárdez, Ó.; Afonso, C. Identifying Real Estate Opportunities Using Machine Learning. Appl. Sci. 2018, 8, 2321

[17] Park, B., Kwon Bae, J. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2014.11.040

[18] Phan, T. D. (2019). Housing price prediction using machine learning algorithms: The case of Melbourne city, Australia. Proceedings - International Conference on Machine Learning and Data Engineering, ICMLDE 2018, 8–13. https://doi.org/10.1109/iCMLDE.2018.00017

[19] Neloy, A. A., Haque, H. M. S., Ul Islam, M. M. (2019). Ensemble Learning Based Rental Apartment Price Prediction Model by Categorical Features Factoring. https://doi.org/10.1145/3318299.3318377

[20] Morito Tsutsumi, Yasushi Yoshida, Hajime Seya, Yuichiro Kawaguchi (2007) Spatial analysis of Tokyo apartment market. Presented at the first world conference of the spatial econometrics association. Fitzwilliam College, University of Cambridge, 11–14 July 2007

[21] Nakagawa, M., Saito, M., Yamaga, H. (2007). Earthquake risk and housing rents: Evidence from the Tokyo Metropolitan Area. Regional Science and Urban Economics. https://doi.org/10.1016/j.regsciurbeco.2006.06.009

[22] Shimizu C, Yasumoto S, Asami Y, Clark TN (2014) Do urban amenities drive housing rent? Grant-in-Aid for Scientific Research(S) real estate markets, financial crisis, and economic growth: an integrated economic approach working paper series no. 9

[23] Sadayuki, T. (2018). Measuring the spatial effect of multiple sites: An application to housing rent and public transportation in Tokyo, Japan. Regional Science and Urban Economics. https://doi.org/10.1016/j.regsciurbeco.2018.03.002

[24] Environmental Policy Section, General Affairs Division, Bureau of Environment. TOKYO'S ENVIRONMENTAL POLICY. , TOKYO'S ENVIRONMENTAL POLICY.

[25] Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78. https://doi.org/10.1145/2347736.2347755

[26] Geocoding API. Google Maps Platform. Google. https://developers.google.com/maps/documentation/geocoding/intro

[27] Distance Matrix API. Google Maps Platform. Google. https://developers.google.com/maps/documentation/distance-matrix/intro

[28] Tokyo Metropolitan Government. Tokyo Regional Earthquake Risk Survey. Tokyo Bureau of Urban Development. http://www.toshiseibi.metro.tokyo.jp/bosai/chousa_6/home.htm. [Japanese]

[29] Atmospheric Environmental Regional Observation System : AEROS. AEROS soramame. Japan Ministry of the Environment. http://soramame.taiki.go.jp [Japanese]

[30] Parks in Tokyo. Tokyo Bureau of Construction . Tokyo Bureau of Construction. http://www.kensetsu.metro.tokyo.jp/english/jigyo/park/01.html.

[31] Metropolitan Police Department

Criminal Statistics in Tokyo by Administrative District and Ku https://www.keishicho.metro.tokyo.jp/about_mpd/jokyo_tokei/jokyo/ninchikensu.html [Japanese]

[32] Tokyo Metropolitan Government. TOKYO STATISTICAL YEARBOOK. Statistics of Tokyo. Management and Coordination Section, Statistics Division, Bureau of General Affairs. http://www.toukei.metro.tokyo.jp/tnenkan/tn-eindex.htm. Accessed 22 June 2019

[33] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[34] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011)

[35] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

[36] Colaboratory: Frequently Asked Questions, Jun. 2018, [online] Available: https://research.google.com/colaboratory/faq.html.

[37] Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. ACM SIGKDD Explorations Newsletter. https://doi.org/10.1145/507533.507538

[38] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NIPS '17.

[39] Microsoft. Features. LightGBM 2.2.4 documentation. https://lightgbm.readthedocs.io/en/latest/Features.html

[40] Microsoft. (2019, May 8). microsoft/LightGBM. GitHub. https://github.com/microsoft/LightGBM/tree/master/examples

[41] A. Anghel, N. Papandreou, T. P. Parnell, et al. “Benchmarking and Optimization of Gradient Boosted Decision Tree Algorithms”. In: CoRR abs/1809.04559 (Oct. 2018)

[42] Fmfn. (2019, May 14). fmfn/BayesianOptimization. GitHub. https://github.com/fmfn/BayesianOptimization