Introduction

The short-term rental market is highly competitive, and understanding what makes an Airbnb listing successful is crucial for hosts and investors. Factors such as location, price, availability, host details, and guest engagement metrics can significantly impact a property's demand. This project aims to analyze Airbnb listings in New York City (AB_NYC_2019 dataset) and develop predictive models to determine key drivers of listing performance, including pricing strategies, review impact, and demand segmentation.

Table of Contents

Section 1: Import Libraries

In [33]:
```
import os
import time
import uuid
from datetime import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS
import folium
from folium.plugins import MarkerCluster, FastMarkerCluster, HeatMap
from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelEncode
from sklearn.compose import ColumnTransformer, make_column_transformer
from sklearn.model_selection import train_test_split, cross_val_score, GridS
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressc
from xgboost import XGBRegressor, XGBClassifier
from lightgbm import LGBMRegressor
from catboost import CatBoostRegressor
from sklearn.pipeline import Pipeline
from sklearn.cluster import KMeans
```

```
from sklearn.decomposition import PCA
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_scor
from sklearn.model_selection import learning_curve
```

Section 2: Initial Data Exploration & Cleaning

In [34]:
```
if not os.path.exists('figures'):
    os.makedirs('figures')
```

In [35]:
```
df_ab = pd.read_csv('AB_NYC_2019.csv')
```

In [36]:
```
df_ab.head()
```

Out[36]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem

In [37]:
```
df_ab.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

In [38]:
```
# Drop irrelevant columns
df_ab = df_ab.drop(['id', 'host_id', 'host_name'], axis=1)
```

In [39]:
```
# Handle missing values
df_ab.loc[df_ab['number_of_reviews'] == 0, 'reviews_per_month'] = 0
df_ab['has_review'] = df_ab['last_review'].notna().astype(int)
df_ab['last_review'] = pd.to_datetime(df_ab['last_review'], errors='coerce')
df_ab['days_since_last_review'] = (pd.to_datetime('2025-04-22') - df_ab['las
df_ab.loc[df_ab['has_review'] == 0, 'days_since_last_review'] = 0
df_ab = df_ab.drop(['last_review'], axis=1)
df_ab = df_ab.dropna(subset=['name', 'neighbourhood_group', 'neighbourhood',
```

In [40]:
```
# Remove outliers (price <= $1000, price >= $10, reasonable minimum_nights)
df_ab = df_ab[(df_ab['price'] <= 1000) & (df_ab['price'] >= 10)]
df_ab = df_ab[df_ab['minimum_nights'] <= 365]
df_ab = df_ab[df_ab['availability_365'] <= 365]
```

In [41]:
```
# Log-transform price to reduce skewness
df_ab['log_price'] = np.log1p(df_ab['price'])
```

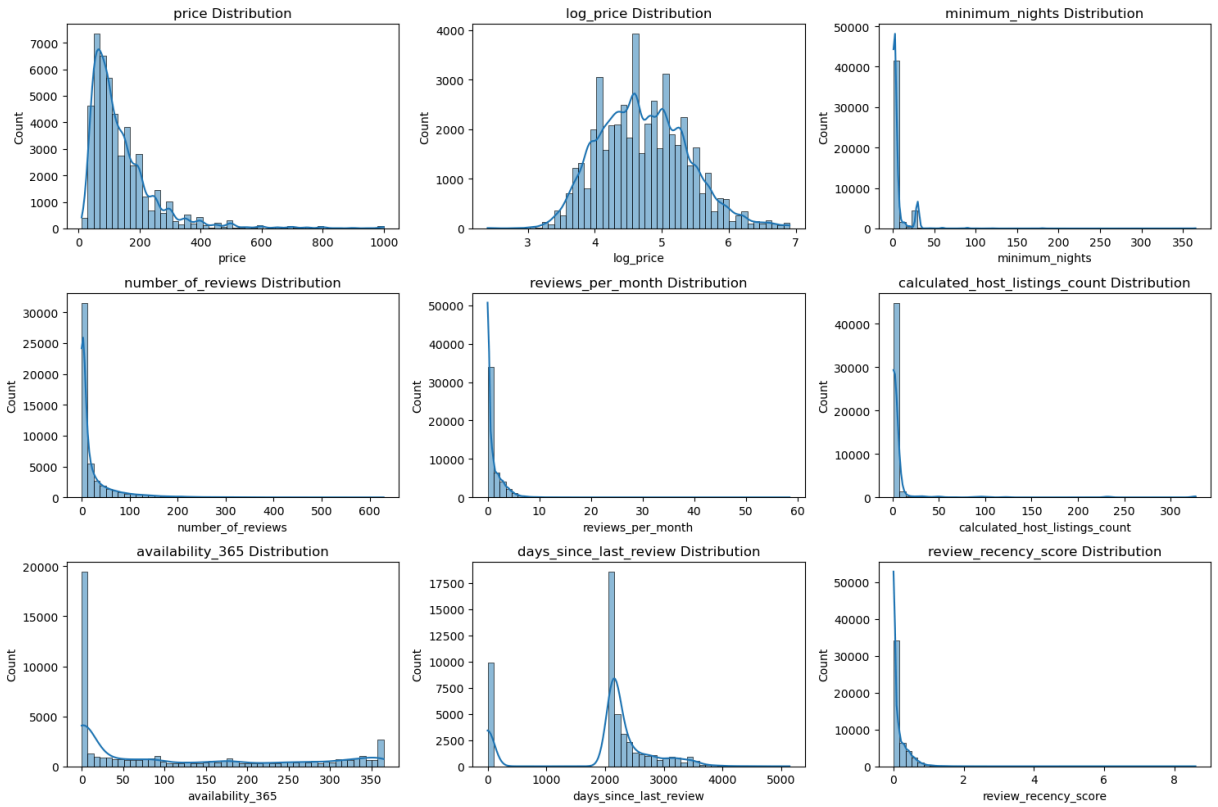
In [42]:
```
# Feature engineering: Review recency score
df_ab['review_recency_score'] = df_ab['reviews_per_month'] * (1 / (1 + df_ab
```

In [43]:
```
# Save cleaned dataset
df_ab.to_csv('cleaned_AB_NYC_2019.csv', index=False)
print("Cleaned dataset saved as 'cleaned_AB_NYC_2019.csv'")
```

```
Cleaned dataset saved as 'cleaned_AB_NYC_2019.csv'
```

Section 3: EDA

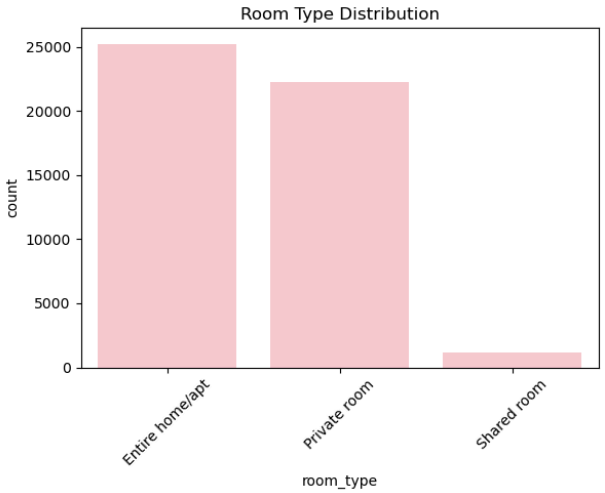
In [44]:
```
# Feature Distribution: Numerical Features
numerical_features = ['price', 'log_price', 'minimum_nights', 'number_of_rev
                      'reviews_per_month', 'calculated_host_listings_count',
                      'availability_365', 'days_since_last_review', 'review_
plt.figure(figsize=(15, 10))
for i, feature in enumerate(numerical_features, 1):
    plt.subplot(3, 3, i)
    sns.histplot(df_ab[feature], bins=50, kde=True)
    plt.title(f'{feature} Distribution')
plt.tight_layout()
plt.savefig('figures/numerical_distributions.png')
plt.show()
plt.close()
```

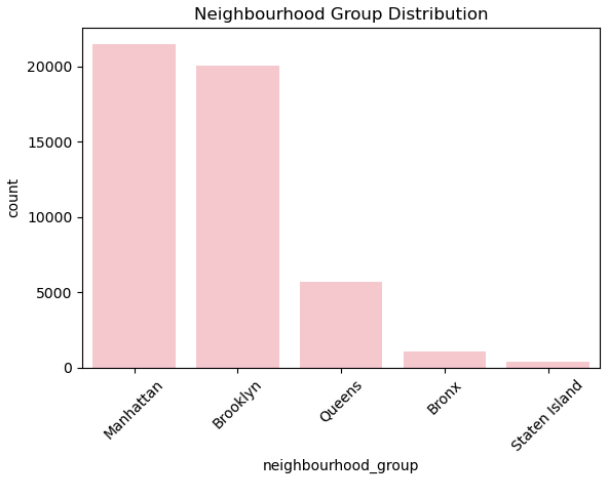


The plots show that most Airbnb listings have low prices, few reviews, short minimum stays, and limited availability, with strong right-skewed distributions. Log-transforming price normalizes its distribution. Peaks in availability_365 (e.g., 0 and 365) and days_since_last_review suggest inactive or default settings. Overall, the data is skewed with many low values and a few extreme outliers, highlighting the need for normalization in analysis.

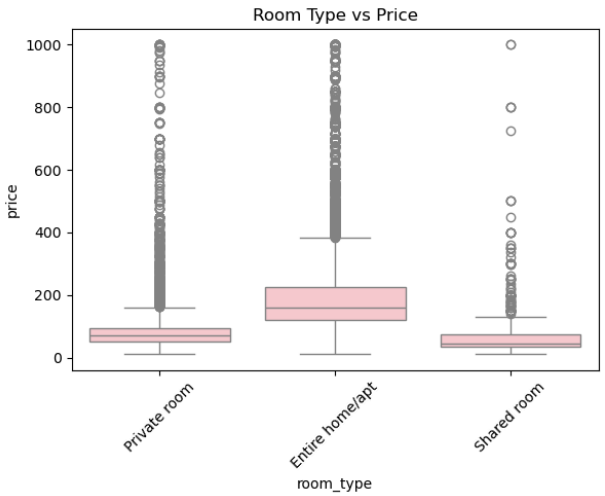
In [45]:
```
# Categorical Features Distribution
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.countplot(data=df_ab, x='room_type', order=df_ab['room_type'].value_cour
plt.title('Room Type Distribution')
plt.xticks(rotation=45)
plt.subplot(1, 2, 2)
```

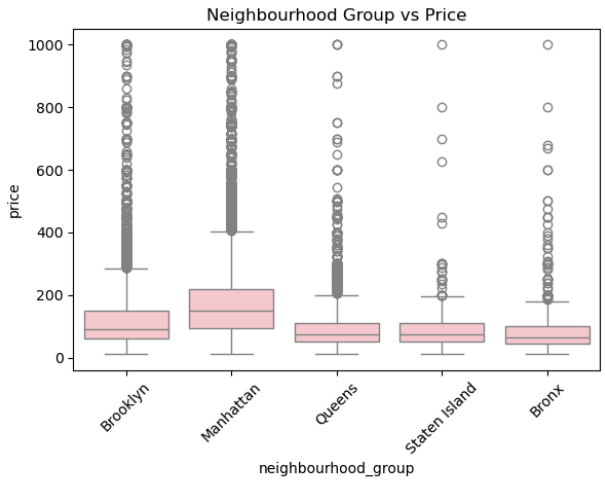
```
sns.countplot(data=df_ab, x='neighbourhood_group', order=df_ab['neighbourhoo
plt.title('Neighbourhood Group Distribution')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('figures/categorical_distributions.png')
plt.show()
plt.close()
```





In [46]:
```
# Price vs. Categorical Features
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.boxplot(data=df_ab, x='room_type', y='price', color='pink')
plt.title('Room Type vs Price')
plt.xticks(rotation=45)
plt.subplot(1, 2, 2)
sns.boxplot(data=df_ab, x='neighbourhood_group', y='price', color='pink')
plt.title('Neighbourhood Group vs Price')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('figures/price_vs_categorical.png')
plt.show()
plt.close()
```

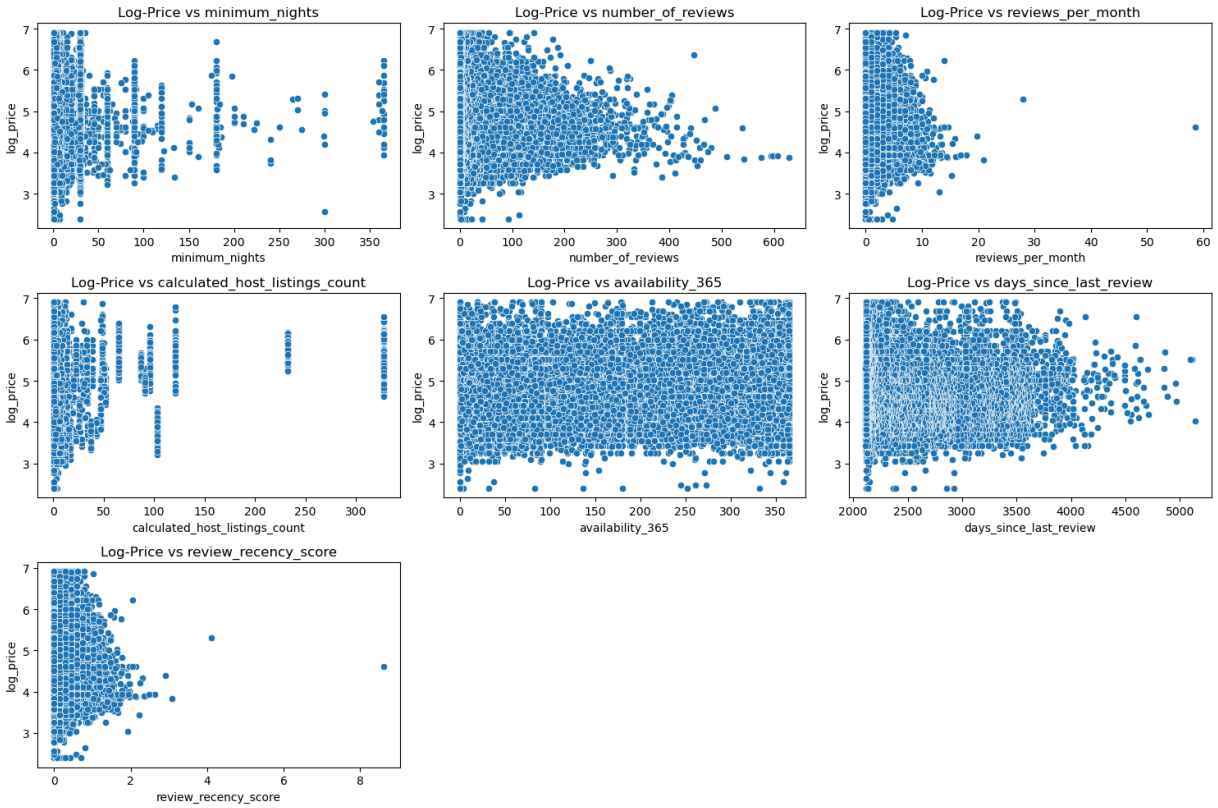




The boxplots show that entire homes/apartments tend to have the highest prices, while shared rooms are the cheapest. Manhattan has the highest price range among

neighborhoods, followed by Brooklyn. The bar charts reveal that entire homes/apartments and private rooms are the most common room types, and most listings are concentrated in Manhattan and Brooklyn.

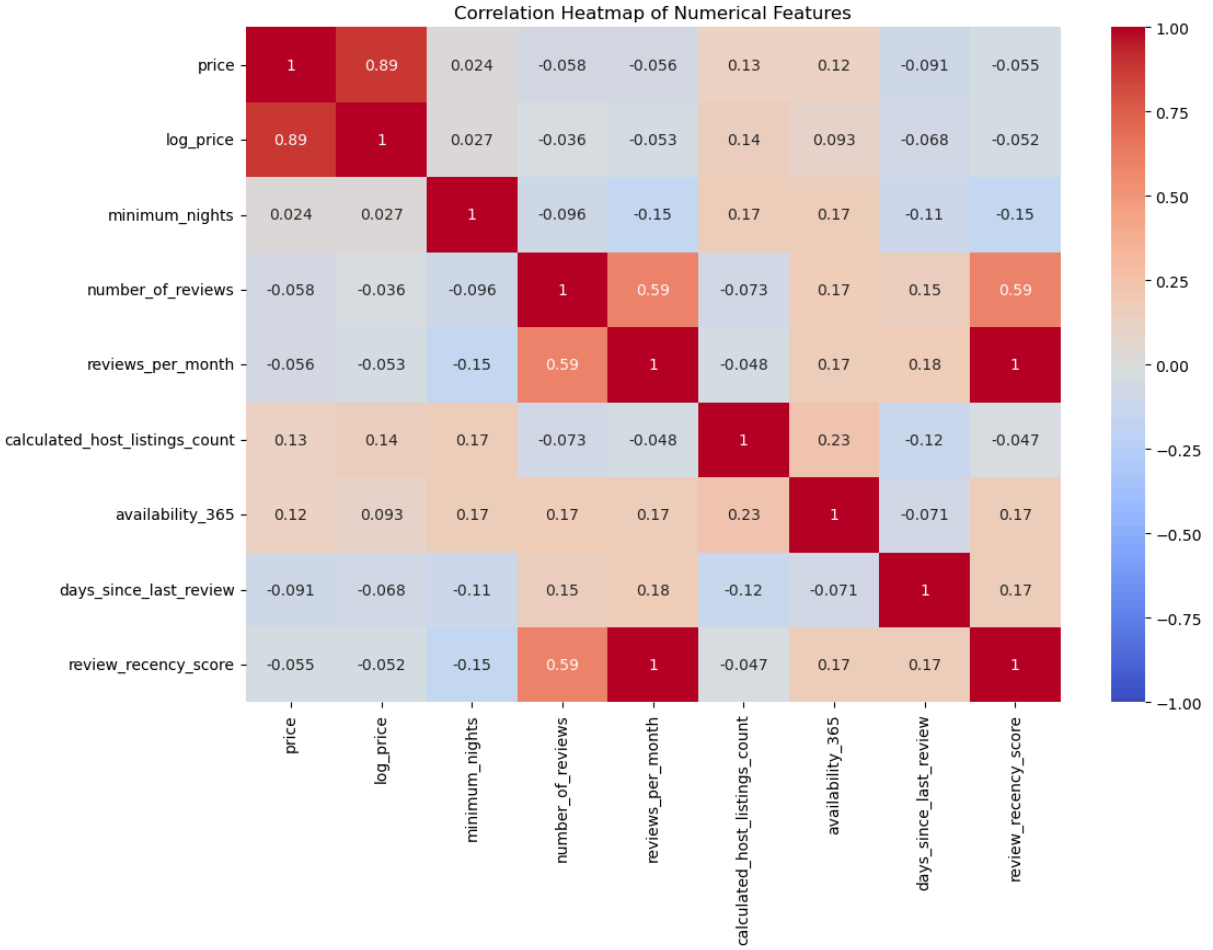
In [47]:
```
# Numerical Features vs. Log-Price
plt.figure(figsize=(15, 10))
for i, feature in enumerate(numerical_features[2:], 1):  # Exclude price and
    plt.subplot(3, 3, i)
    sns.scatterplot(data=df_ab[df_ab[feature] != 0] if feature == 'days_sinc
                    x=feature, y='log_price')
    plt.title(f'Log-Price vs {feature}')
plt.tight_layout()
plt.savefig('figures/numerical_vs_logprice.png')
plt.show()
plt.close()
```



These scatter plots show the relationship between log_price and several variables. Most features, like minimum_nights, number_of_reviews, and reviews_per_month, show weak or no clear linear correlation with price. Listings with higher prices are scattered across all ranges of availability and host activity metrics (availability_365, calculated_host_listings_count, days_since_last_review). Overall, log_price seems relatively independent of these individual features, suggesting other factors (e.g., location, room type) may play a stronger role in pricing.

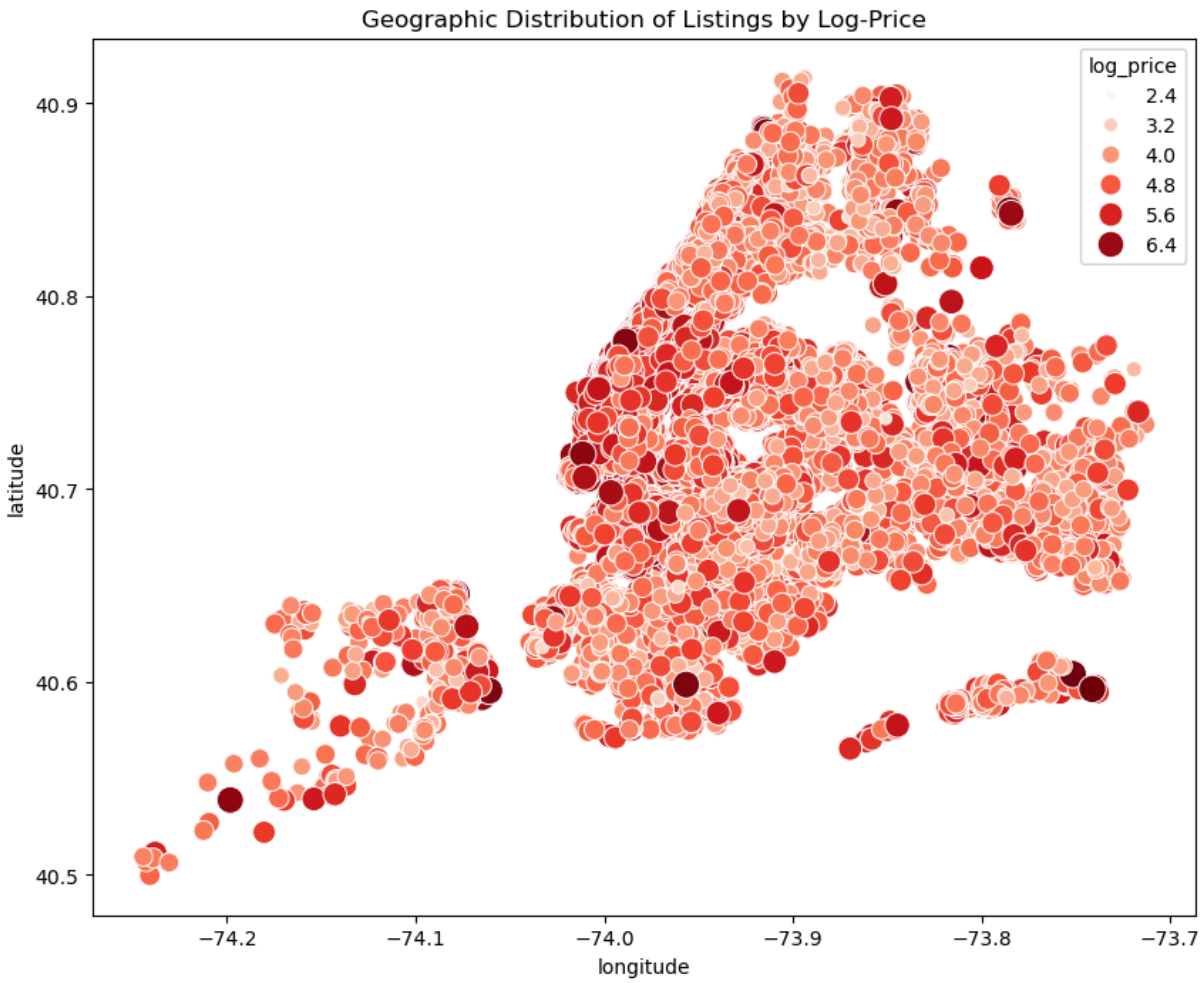
In [48]:
```
# Correlation Heatmap
plt.figure(figsize=(12, 8))
corr_matrix = df_ab[numerical_features].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
```

```
plt.title('Correlation Heatmap of Numerical Features')
plt.savefig('figures/correlation_heatmap.png')
plt.show()
plt.close()
```



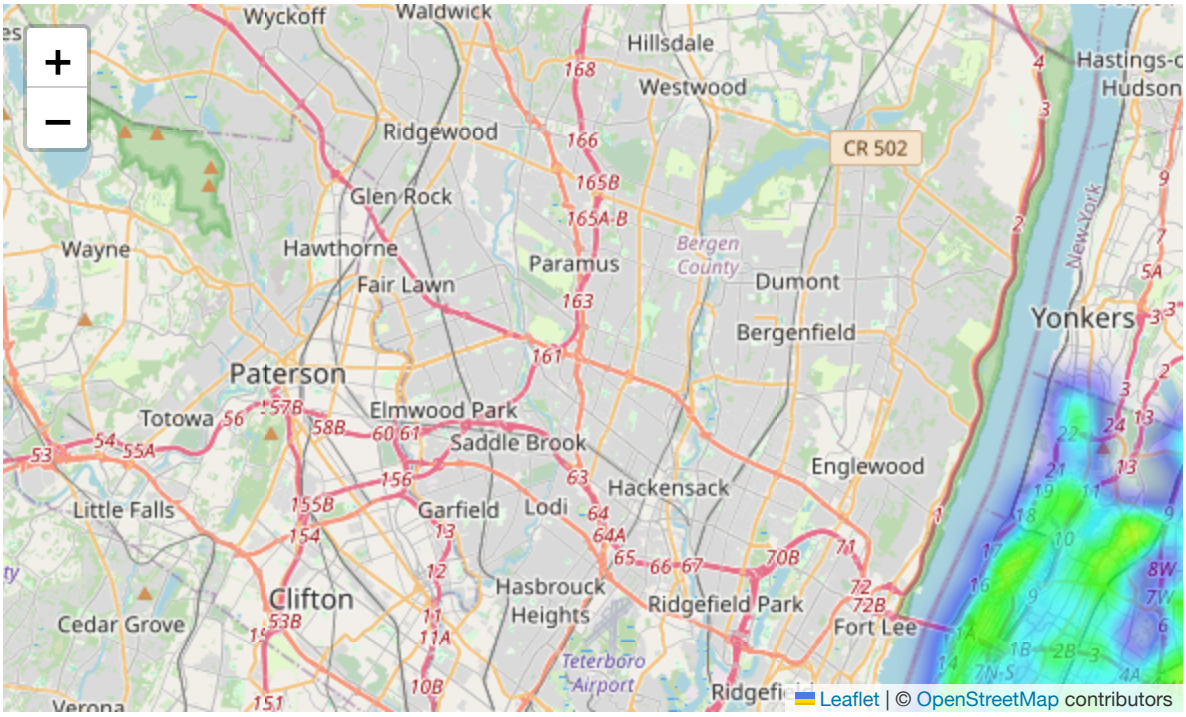
The correlation heatmap reveals that most numerical features exhibit slightly stronger and more consistent correlations with log_price compared to the original price. For instance, features such as calculated_host_listings_count and availability_365 show higher correlation values with log_price, indicating a more stable relationship. Moreover, log_price mitigates the influence of extreme values present in price, making it a more suitable target variable for predictive modeling. Based on these observations, we chose to use log_price as the dependent variable in our machine learning models to improve performance and interpretability.

In [49]:
```
# Location vs. Price: Geographic Visualization
plt.figure(figsize=(10, 8))
sns.scatterplot(data=df_ab, x='longitude', y='latitude', hue='log_price', si
plt.title('Geographic Distribution of Listings by Log-Price')
plt.savefig('figures/geo_scatter_logprice.png')
plt.show()
plt.close()
```

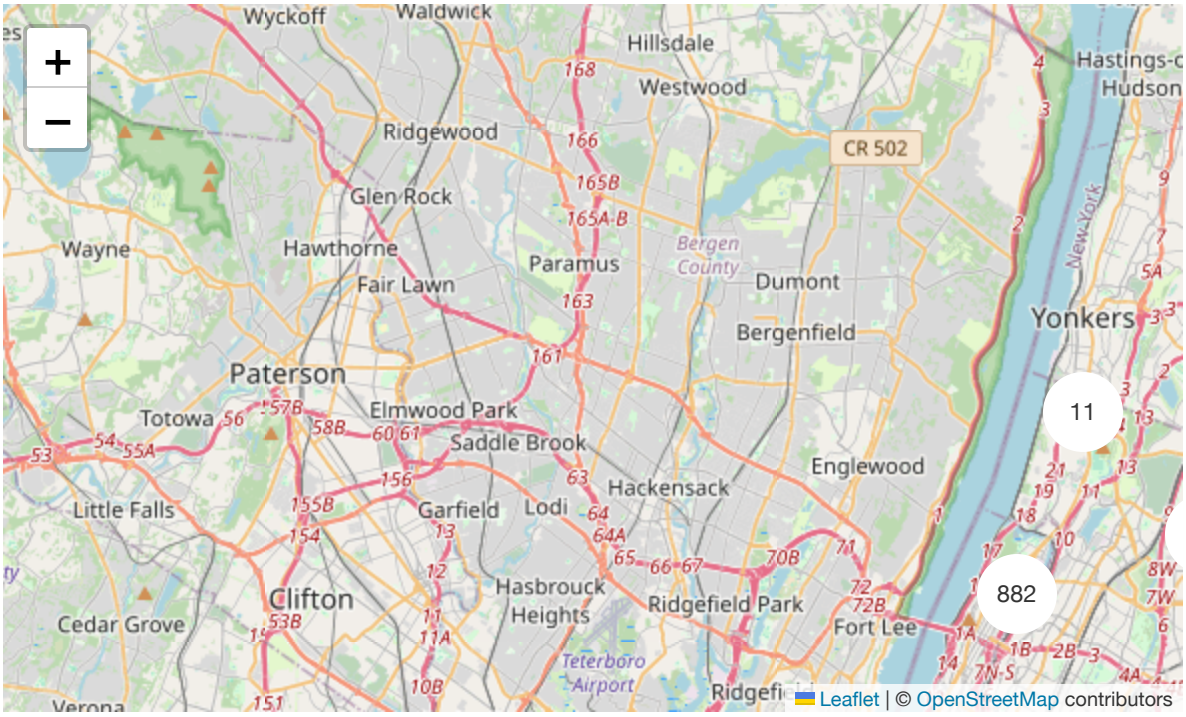


This geographic distribution plot shows Airbnb listings across New York City, colored by their log_price. Darker red dots represent higher log-transformed prices. We observe that higher-priced listings tend to cluster in central and desirable areas such as lower Manhattan and parts of Brooklyn, while lighter-colored (lower-priced) listings are more spread out in outer boroughs. This spatial trend highlights the influence of location on listing prices and suggests that geographic features will be important in modeling and predicting price.

In [50]:
```
# Price Intensity Heatmap
# shows the geographic distribution of listing prices using color gradients
m = folium.Map(location=[40.80, -73.80], zoom_start=11)
heat_data = [[row['latitude'], row['longitude'], row['price']] for idx, row
HeatMap(heat_data, radius=10).add_to(m)
m.save('figures/price_heatmap.html')
display(m)
```



In [51]:
```
# Listing Density Cluster Map
# displays the number of listings per area using marker clusters, allowing u
Lat, Long = 40.80, -73.80
m_2 = folium.Map(location=[Lat, Long], zoom_start=11)

marker_cluster = MarkerCluster(name='Density Cluster', control=False).add_to

for idx, row in df_ab.iterrows():
    popup_text = f"""
    <b>Price:</b> ${row['price']}
    """
    folium.Marker(
        location=[row['latitude'], row['longitude']],
        popup=folium.Popup(popup_text, max_width=300),
        icon=folium.Icon(color='blue', icon='home', prefix='fa')
    ).add_to(marker_cluster)

m_2.save('figures/density_cluster_map.html')
display(m_2)
```



In [52]:
```
# Word Cloud for High and Low-Priced Listings
df_high = df_ab[df_ab['price'] > df_ab['price'].quantile(0.75)]
df_low = df_ab[df_ab['price'] < df_ab['price'].quantile(0.25)]
text_high = " ".join(str(name) for name in df_high['name'])
text_low = " ".join(str(name) for name in df_low['name'])

wordcloud_high = WordCloud(width=800, height=400, background_color="white",
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_high, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud: High-Priced Listings")
plt.savefig('figures/wordcloud_high.png')
plt.show()
plt.close()
```

In [53]:
```
wordcloud_low = WordCloud(width=800, height=400, background_color="white", s
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_low, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud: Low-Priced Listings")
plt.savefig('figures/wordcloud_low.png')
plt.show()
plt.close()
```

In [54]:
```
# Text Analysis: TF-IDF on Listing Names
tfidf = TfidfVectorizer(max_features=100, stop_words='english')
tfidf_matrix = tfidf.fit_transform(df_ab['name'].dropna())
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=tfidf.get_feature_na
tfidf_price_corr = tfidf_df.corrwith(df_ab['price']).sort_values(ascending=F
```

```
print("Top 10 TF-IDF Features Correlated with Price:")
print(tfidf_price_corr)
```

```
Top 10 TF-IDF Features Correlated with Price:
2br         0.014250
chelsea     0.011204
loft        0.010082
near        0.009404
subway      0.009190
queens      0.008538
modern      0.008328
duplex      0.007618
gorgeous    0.007474
nyc         0.007163
dtype: float64
```

The text analysis provides insights into how listing titles relate to price. The word clouds show that high-priced listings often include words like "Luxury," "Spacious," "Modern," and "Apartment," along with location names like "Manhattan" and "Williamsburg", emphasizing size, quality, and premium areas. In contrast, low-priced listings frequently use terms like "Private Room," "Cozy," and "Bushwick," suggesting smaller, shared spaces in more affordable neighborhoods. The TF-IDF correlation analysis further supports this: words like "2br," "chelsea," "loft," and "modern" are among the top terms most positively correlated with higher prices. These findings suggest that both the descriptive language and location cues in listing titles can be strong indicators of price.

Section 4: Preprocessing

In [55]:
```
categorical_features = ['neighbourhood_group', 'room_type']
numeric_features = ['minimum_nights', 'number_of_reviews', 'reviews_per_mont
                    'calculated_host_listings_count', 'availability_365',
                    'has_review', 'days_since_last_review', 'review_recency_
passthrough_features = ['latitude', 'longitude']
target = 'log_price'  # Use log_price to reduce skewness
```

In [56]:
```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features
        ('pass', 'passthrough', passthrough_features)
    ])

X = df_ab[numeric_features + categorical_features + passthrough_features]
y = df_ab[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rar
```

Section 5: Machine Learning

In [57]:
```
models = [
    ('LinearRegression', LinearRegression()),
    ('Ridge', Ridge()),
    ('Lasso', Lasso()),
    ('KNN', KNeighborsRegressor()),
    ('RandomForest', RandomForestRegressor(random_state=42)),
    ('GradientBoosting', GradientBoostingRegressor(random_state=42)),
    ('XGBoost', XGBRegressor(random_state=42, verbosity=0)),
    ('LightGBM', LGBMRegressor(random_state=42)),
    ('CatBoost', CatBoostRegressor(random_state=42, verbose=0))
]
```

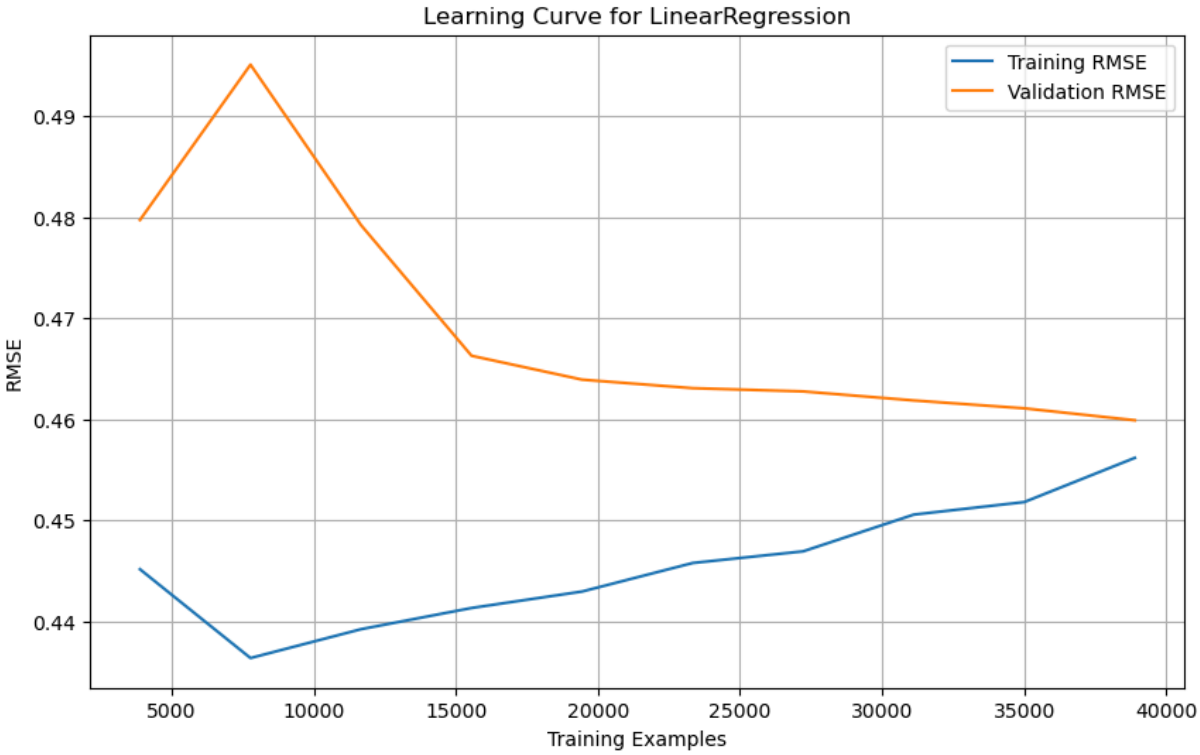
In [58]:
```
# Hyperparameter grids
param_grids = {
    'LinearRegression': {},
    'Ridge': {'regressor__alpha': [0.1, 1.0, 10.0]},
    'Lasso': {'regressor__alpha': [0.1, 1.0, 10.0]},
    'KNN': {'regressor__n_neighbors': [3, 5, 7]},
    'RandomForest': {'regressor__n_estimators': [50, 100], 'regressor__max_d
    'GradientBoosting': {'regressor__n_estimators': [50, 100], 'regressor__l
    'XGBoost': {'regressor__n_estimators': [50, 100], 'regressor__learning_r
    'LightGBM': {'regressor__n_estimators': [50, 100], 'regressor__learning_
    'CatBoost': {'regressor__iterations': [50, 100], 'regressor__learning_ra
}
```

In [59]:
```
results = []

# Function to plot learning curves
def plot_learning_curve(estimator, title, X, y, cv=5, train_sizes=np.linspac
    plt.figure(figsize=(10, 6))
    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=cv, n_jobs=-1, train_sizes=train_sizes, scoring=
    )
    train_scores_mean = np.sqrt(-train_scores.mean(axis=1))
    test_scores_mean = np.sqrt(-test_scores.mean(axis=1))

    plt.plot(train_sizes, train_scores_mean, label='Training RMSE')
    plt.plot(train_sizes, test_scores_mean, label='Validation RMSE')
    plt.title(title)
    plt.xlabel('Training Examples')
    plt.ylabel('RMSE')
    plt.legend(loc='best')
    plt.grid(True)
    plt.savefig(f'figures/learning_curve_{title.lower().replace(" ", "_")}.p
    plt.show()
    plt.close()
```

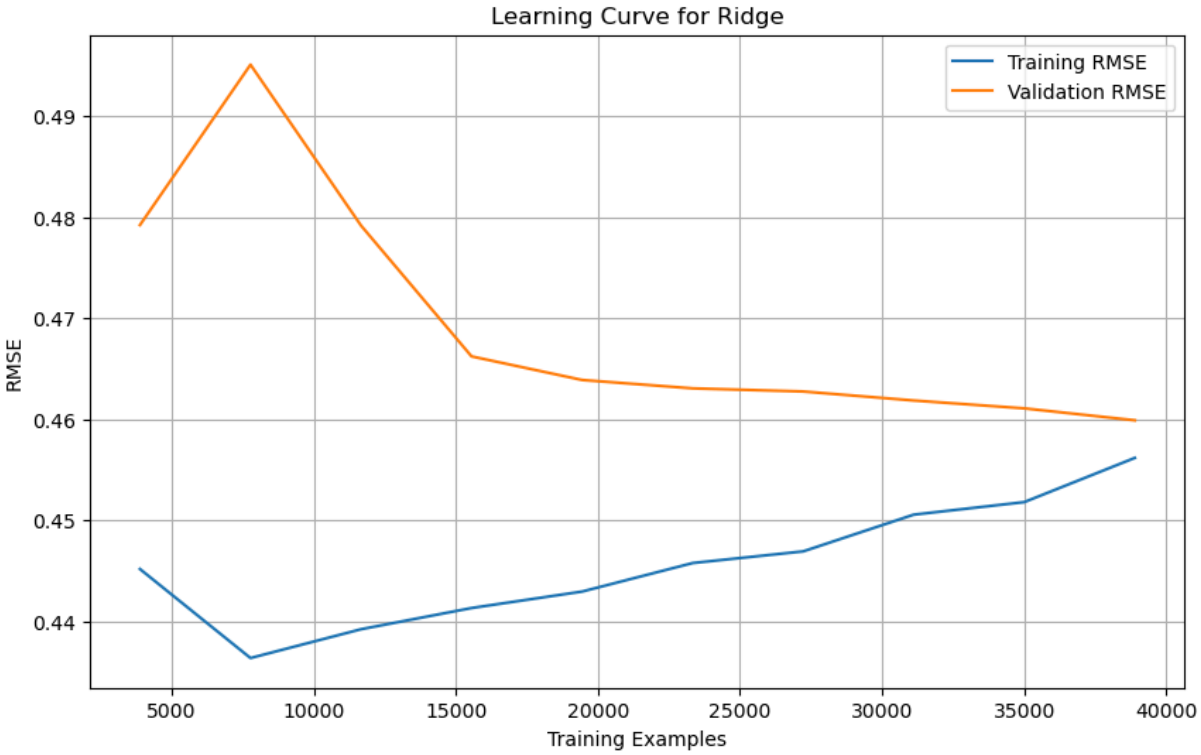
In [60]:
```
for name, model in models:
    print(f"Training: {name}")
    start_time = time.time()

    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('regressor', model)
```

```
    ])

    if param_grids[name]:
        grid = GridSearchCV(pipeline, param_grids[name], cv=5, n_jobs=-1, sc
        grid.fit(X_train, y_train)
        best_model = grid.best_estimator_
        print("Best Params:", grid.best_params_)
    else:
        best_model = pipeline.fit(X_train, y_train)

    y_pred = best_model.predict(X_test)

    # Metrics
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    r2 = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    mape = mean_absolute_percentage_error(y_test, y_pred) * 100  # Convert t

    # Cross-validation score
    cv_scores = cross_val_score(best_model, X, y, cv=5, scoring='neg_mean_sc
    cv_rmse = np.sqrt(-cv_scores.mean())

    results.append({
        'Model': name,
        'RMSE': rmse,
        'R²': r2,
        'MAE': mae,
        'MSE': mse,
        'MAPE (%)': mape,
        'CV_RMSE': cv_rmse,
        'Time': time.time() - start_time
    })

    print(f"RMSE: {rmse:.4f}")
    print(f"R²: {r2:.4f}")
    print(f"MAE: {mae:.4f}")
    print(f"MSE: {mse:.4f}")
    print(f"MAPE: {mape:.2f}%")
    print(f"CV RMSE: {cv_rmse:.4f}")
    print(f"Time: {results[-1]['Time']:.2f} sec\n")

    # Plot learning curve
    plot_learning_curve(best_model, f"Learning Curve for {name}", X, y)

# Convert results to DataFrame
results_df = pd.DataFrame(results)

# Display formatted results table
print("\nModel Performance Summary:")
display(results_df.round(4))

# Save results to CSV
results_df.to_csv('model_performance_summary.csv', index=False)
print("Model performance summary saved as 'model_performance_summary.csv'")
```

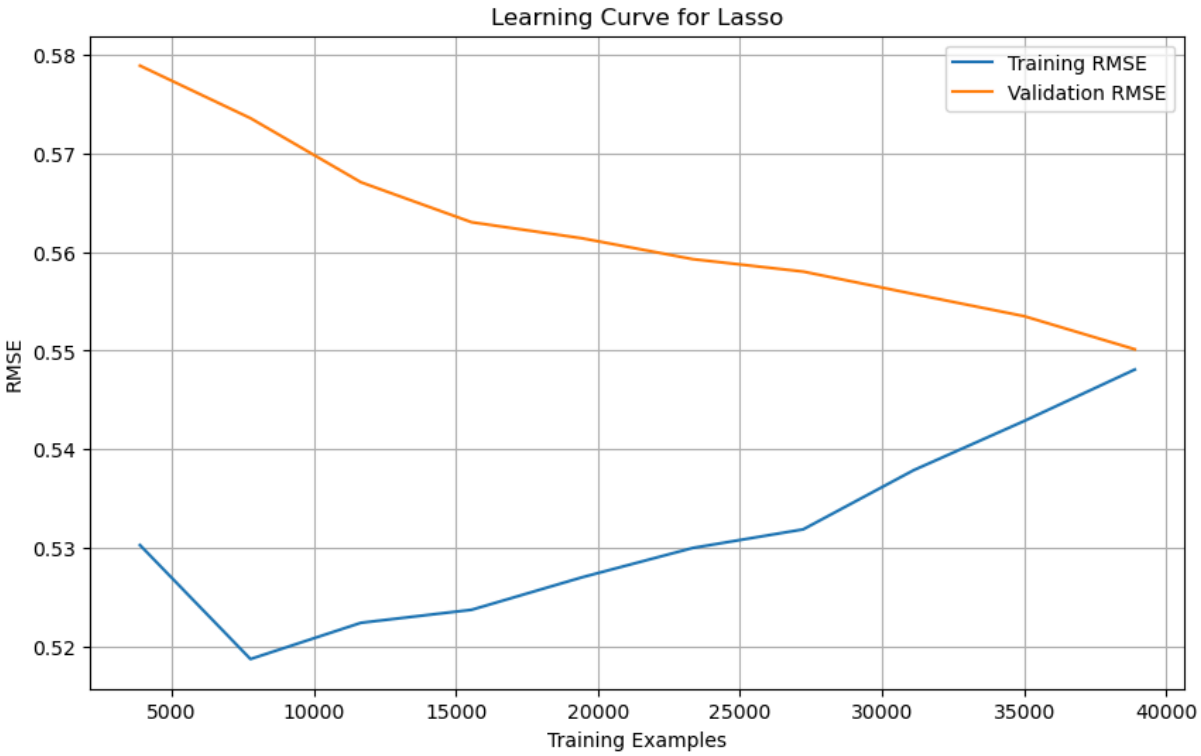
```
Training: LinearRegression
RMSE: 0.4532
R²: 0.5292
MAE: 0.3414
MSE: 0.2054
MAPE: 7.24%
CV RMSE: 0.4599
Time: 0.26 sec
```



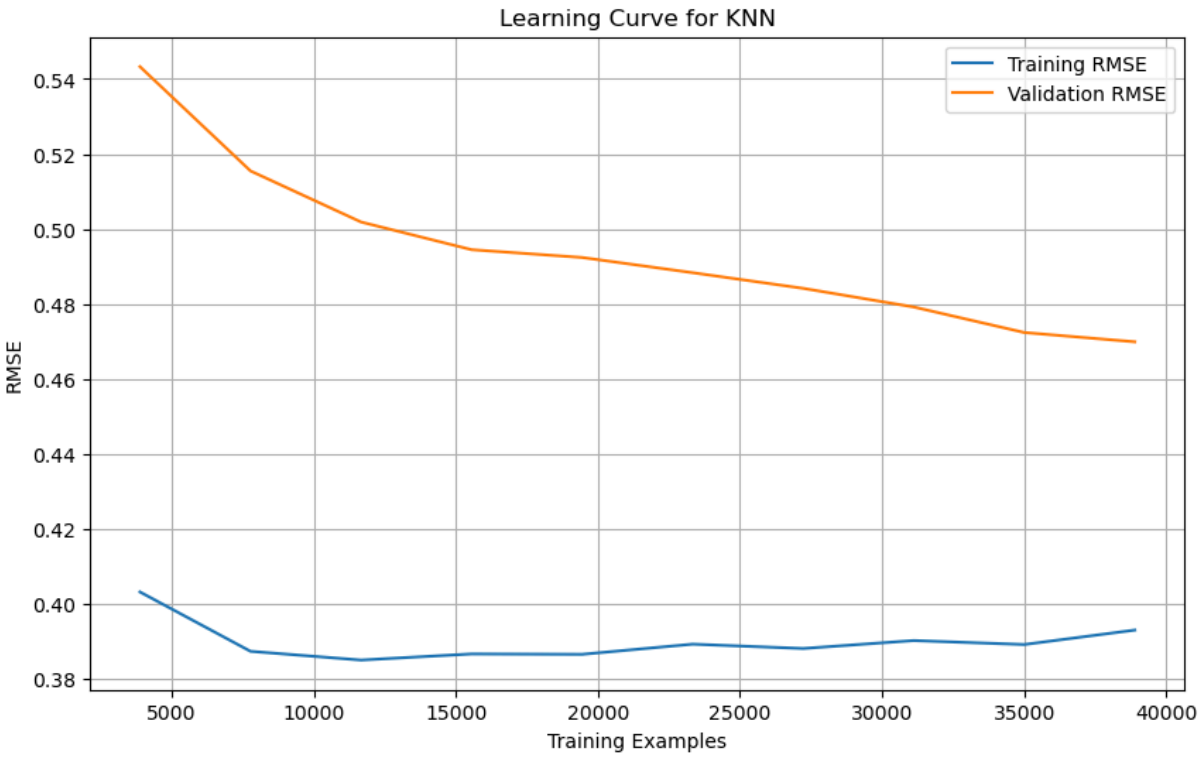

```
Training: Ridge
Best Params: {'regressor__alpha': 0.1}
RMSE: 0.4532
R²: 0.5293
MAE: 0.3414
MSE: 0.2054
MAPE: 7.24%
CV RMSE: 0.4599
Time: 0.52 sec
```



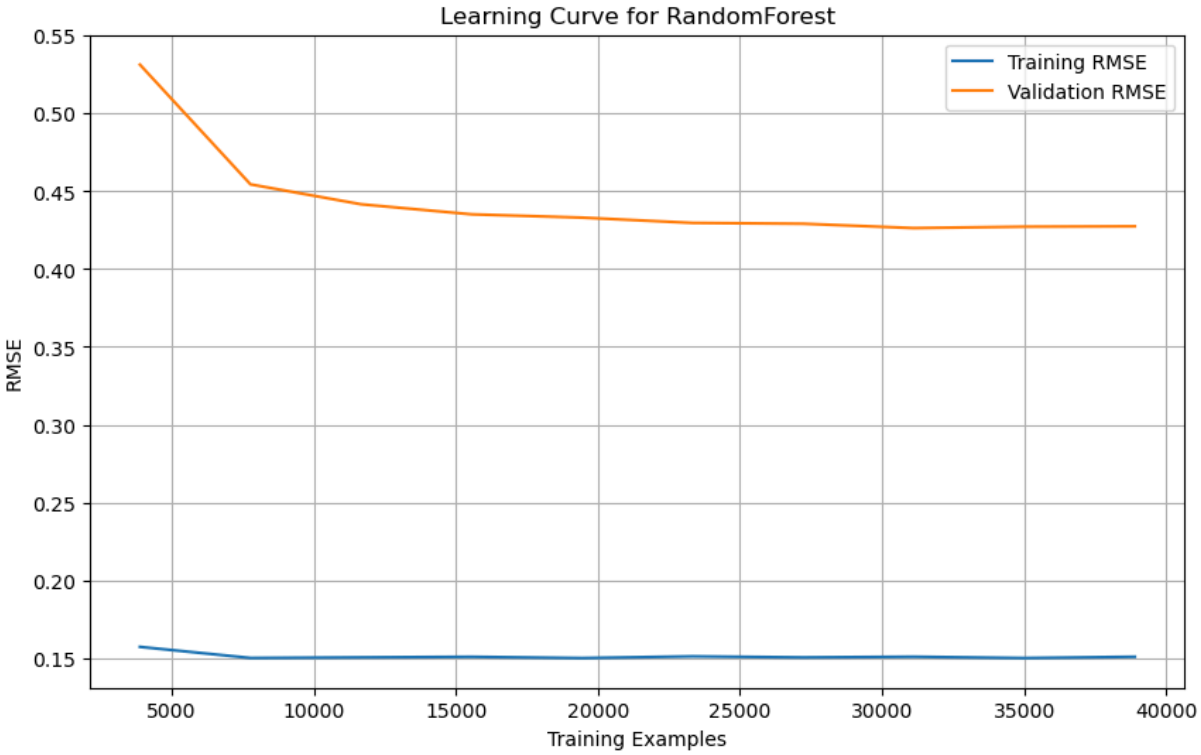

```
Training: Lasso
Best Params: {'regressor__alpha': 0.1}
RMSE: 0.5460
R²: 0.3167
MAE: 0.4246
MSE: 0.2981
MAPE: 9.11%
CV RMSE: 0.5501
Time: 0.44 sec
```



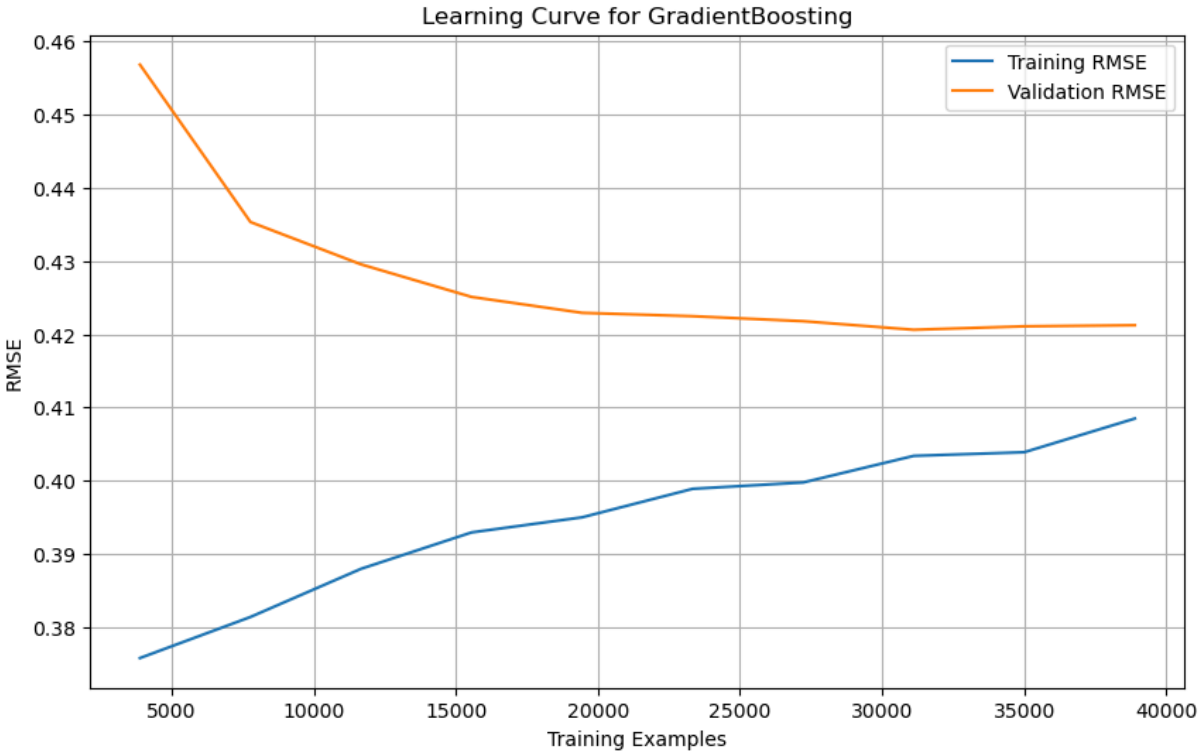
```
Training: KNN
Best Params: {'regressor__n_neighbors': 7}
RMSE: 0.4538
R²: 0.5280
MAE: 0.3458
MSE: 0.2059
MAPE: 7.34%
CV RMSE: 0.4699
Time: 2.64 sec
```



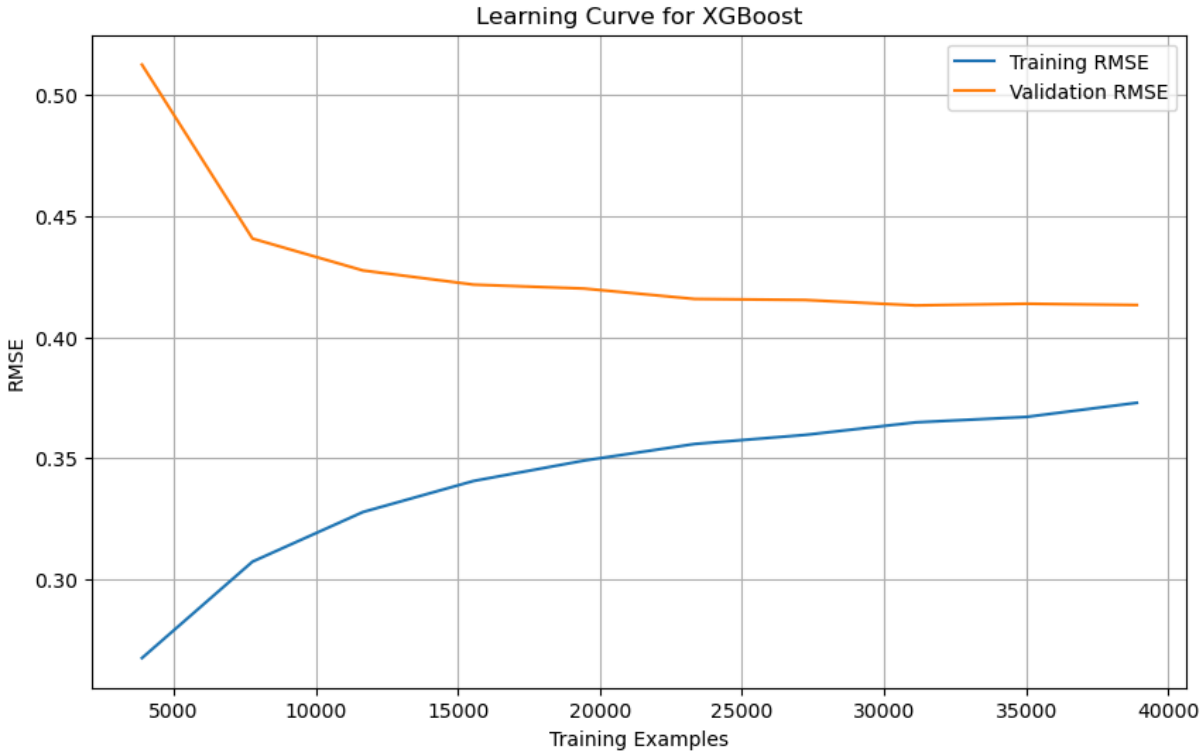

```
Training: RandomForest
Best Params: {'regressor__max_depth': None, 'regressor__n_estimators': 100}
RMSE: 0.4004
R²: 0.6325
MAE: 0.2949
MSE: 0.1603
MAPE: 6.23%
CV RMSE: 0.4275
Time: 158.06 sec
```




```
Training: GradientBoosting
Best Params: {'regressor__learning_rate': 0.1, 'regressor__n_estimators': 10
0}
RMSE: 0.4104
R²: 0.6139
MAE: 0.3055
MSE: 0.1685
MAPE: 6.44%
CV RMSE: 0.4211
Time: 46.60 sec
```




```
Training: XGBoost
Best Params: {'regressor__learning_rate': 0.1, 'regressor__n_estimators': 10
0}
RMSE: 0.3983
R²: 0.6364
MAE: 0.2954
MSE: 0.1586
MAPE: 6.23%
CV RMSE: 0.4141
Time: 2.47 sec
```


Learning Curve for XGBoost
Training RMSE
Validation RMSE
RMSE
0.50
0.45
0.40
0.35
0.30
5000
10000
15000
20000
25000
30000
35000
40000
Training Examples

```
Training: LightGBM
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.009733 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.007859 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.009496 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1905
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.005026 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1907
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.005975 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.002949 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.003040 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1905
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.724360
[LightGBM] [Info] Start training from score 4.721756
[LightGBM] [Info] Start training from score 4.724360
[LightGBM] [Info] Start training from score 4.721908
[LightGBM] [Info] Start training from score 4.718999
[LightGBM] [Info] Start training from score 4.725888
[LightGBM] [Info] Start training from score 4.721756
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002628 seconds.
You can set `force_col_wise=true` to remove the overhead.
```

```
[LightGBM] [Info] Total Bins 1907
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.721908
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001383 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.718999
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002389 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.725888
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.003410 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1905
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.721756
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003152 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003946 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1907
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.721908
[LightGBM] [Info] Start training from score 4.724360
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002376 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.725888
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.002565 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.718999
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002691 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1905
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.721756
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002062 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1907
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.721908
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002464 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.724360
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.001296 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.725888
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003239 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 31114, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.718999
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000432 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1910
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.722582
Best Params: {'regressor__learning_rate': 0.1, 'regressor__n_estimators': 10
0}
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000322 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.704019
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000384 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1912
[LightGBM] [Info] Number of data points in the train set: 38893, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.727057
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000318 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1911
[LightGBM] [Info] Number of data points in the train set: 38893, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.743833
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000313 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1908
[LightGBM] [Info] Number of data points in the train set: 38893, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.732349
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000423 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 38893, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.711489
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
RMSE: 0.3975
R²: 0.6378
MAE: 0.2954
MSE: 0.1580
MAPE: 6.23%
CV RMSE: 0.4141
Time: 12.64 sec

[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000565 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1777
[LightGBM] [Info] Number of data points in the train set: 3889, number of us
ed features: 17
[LightGBM] [Info] Start training from score 4.729235
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000389 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1854
[LightGBM] [Info] Number of data points in the train set: 7778, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.711292
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001126 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1871
[LightGBM] [Info] Number of data points in the train set: 11667, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.697248
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.004027 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1881
[LightGBM] [Info] Number of data points in the train set: 15556, number of u
sed features: 18
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.001579 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1884
[LightGBM] [Info] Number of data points in the train set: 19446, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.681157
[LightGBM] [Info] Start training from score 4.676967
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001580 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1888
[LightGBM] [Info] Number of data points in the train set: 23335, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.673886
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001606 seconds.
You can set `force_col_wise=true` to remove the overhead.
```

```
[LightGBM] [Info] Total Bins 1892
[LightGBM] [Info] Number of data points in the train set: 27224, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.672983
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002162 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.679096
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003856 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1905
[LightGBM] [Info] Number of data points in the train set: 35002, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.688043
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002984 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1904
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.704019
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001317 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1853
[LightGBM] [Info] Number of data points in the train set: 3889, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.860378
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001723 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1873
[LightGBM] [Info] Number of data points in the train set: 7778, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.817113
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001107 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1880
[LightGBM] [Info] Number of data points in the train set: 11667, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.774051
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001682 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1891
[LightGBM] [Info] Number of data points in the train set: 15556, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.738794
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002193 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1893
[LightGBM] [Info] Number of data points in the train set: 19446, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.723066
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000656 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1898
[LightGBM] [Info] Number of data points in the train set: 23335, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.712331
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.005334 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1900
[LightGBM] [Info] Number of data points in the train set: 27224, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.705859
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002549 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1908
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.707863
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002226 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1911
[LightGBM] [Info] Number of data points in the train set: 35002, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.713644
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002231 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1912
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.727062
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.004231 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1853
[LightGBM] [Info] Number of data points in the train set: 3889, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.860378
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.004296 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1873
[LightGBM] [Info] Number of data points in the train set: 7778, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.817113
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003312 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1876
[LightGBM] [Info] Number of data points in the train set: 11667, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.787841
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.004113 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1879
[LightGBM] [Info] Number of data points in the train set: 15556, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.770120
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.012170 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1888
[LightGBM] [Info] Number of data points in the train set: 19446, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.756617
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.004909 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1895
[LightGBM] [Info] Number of data points in the train set: 23335, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.740293
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.002386 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1899
[LightGBM] [Info] Number of data points in the train set: 27224, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.729827
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.009247 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1909
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.728835
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003064 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1911
[LightGBM] [Info] Number of data points in the train set: 35002, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.732286
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003534 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1911
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.743839
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.001036 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1853
[LightGBM] [Info] Number of data points in the train set: 3889, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.860378
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000860 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1873
[LightGBM] [Info] Number of data points in the train set: 7778, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.817113
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.007051 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1876
[LightGBM] [Info] Number of data points in the train set: 11667, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.787841
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002686 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1879
[LightGBM] [Info] Number of data points in the train set: 15556, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.770120
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000888 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1888
[LightGBM] [Info] Number of data points in the train set: 19446, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.756617
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003053 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1889
[LightGBM] [Info] Number of data points in the train set: 23335, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.734513
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002635 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1893
[LightGBM] [Info] Number of data points in the train set: 27224, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.723388
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003283 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1900
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.714479
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003147 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1907
[LightGBM] [Info] Number of data points in the train set: 35002, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.719525
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.001833 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1908
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.732355
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

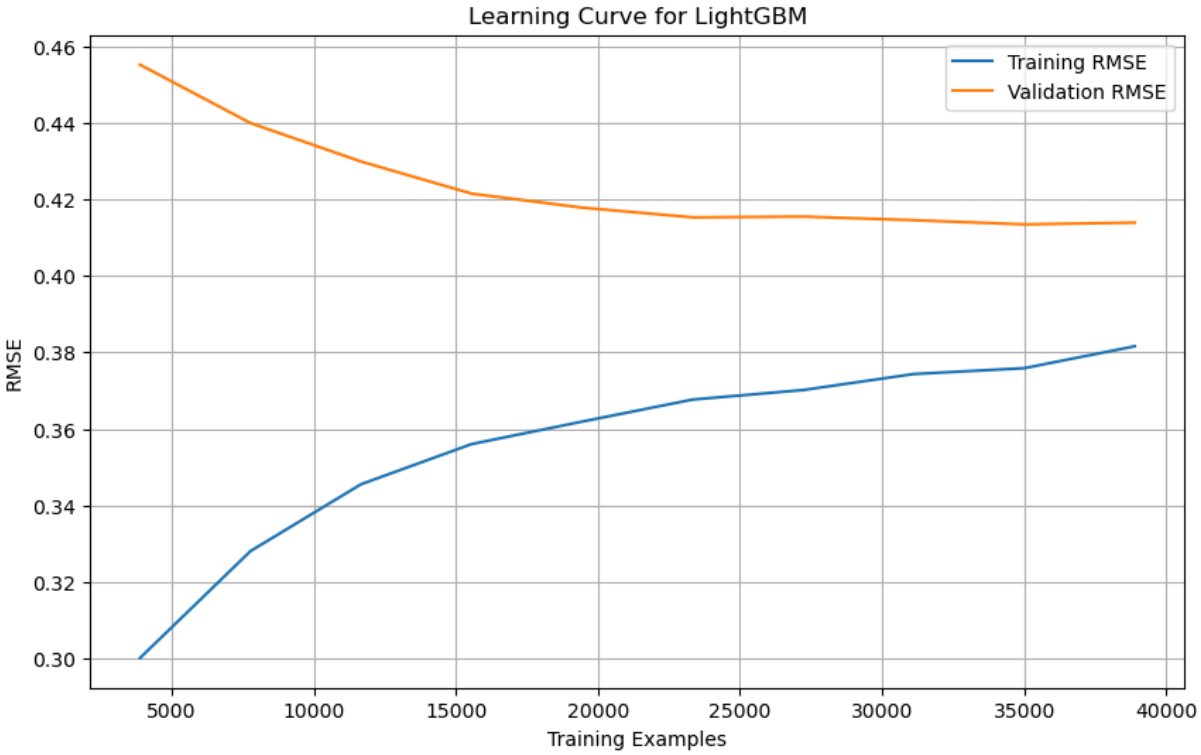
```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002708 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1853
[LightGBM] [Info] Number of data points in the train set: 3889, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.860378
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.000769 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1873
[LightGBM] [Info] Number of data points in the train set: 7778, number of us
ed features: 18
[LightGBM] [Info] Start training from score 4.817113
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.004398 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1876
[LightGBM] [Info] Number of data points in the train set: 11667, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.787841
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.001599 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1879
[LightGBM] [Info] Number of data points in the train set: 15556, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.770120
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003471 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1888
[LightGBM] [Info] Number of data points in the train set: 19446, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.756617
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002941 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1889
[LightGBM] [Info] Number of data points in the train set: 23335, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.734513
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002195 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1893
[LightGBM] [Info] Number of data points in the train set: 27224, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.723388
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.002124 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1897
[LightGBM] [Info] Number of data points in the train set: 31113, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.711506
```

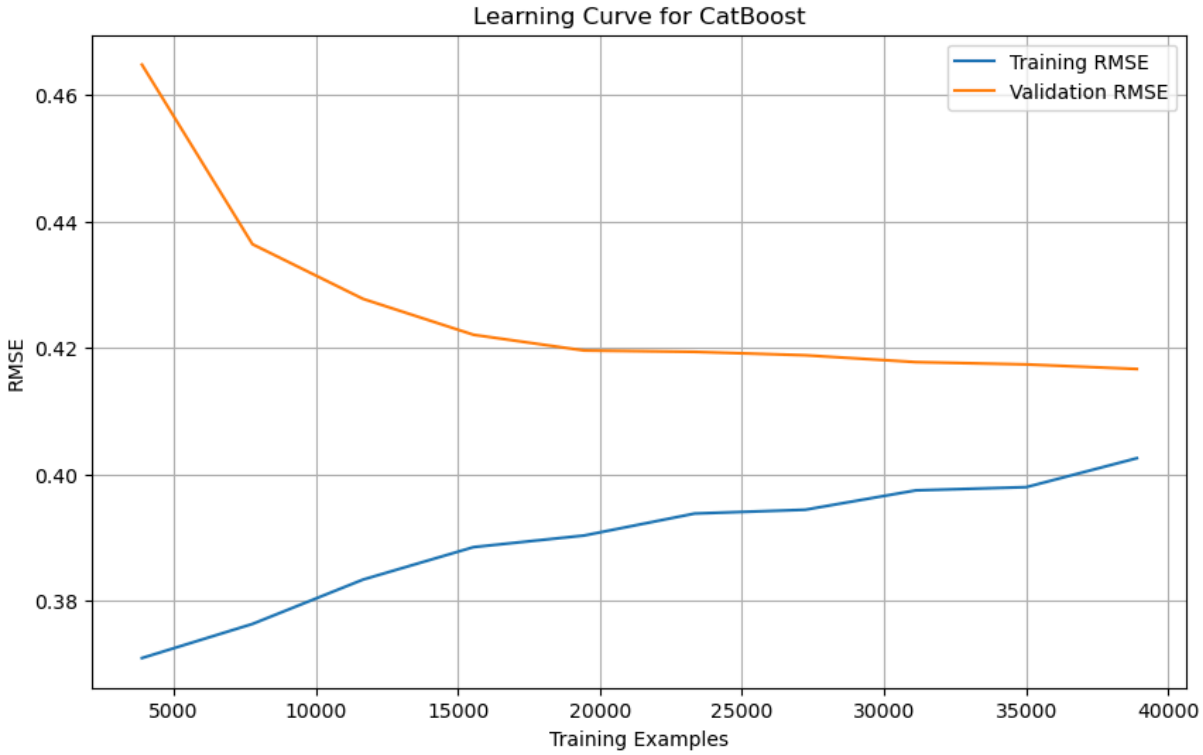
```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.006052 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1902
[LightGBM] [Info] Number of data points in the train set: 35002, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.709979
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of te
sting was 0.003561 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1906
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.711508
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/utils/validation.py:273
9: UserWarning: X does not have valid feature names, but LGBMRegressor was f
itted with feature names
  warnings.warn(
```




```
Training: CatBoost
Best Params: {'regressor__depth': 6, 'regressor__iterations': 100, 'regresso
r__learning_rate': 0.1}
RMSE: 0.4054
R²: 0.6234
MAE: 0.3017
MSE: 0.1643
MAPE: 6.36%
CV RMSE: 0.4177
Time: 8.05 sec
```



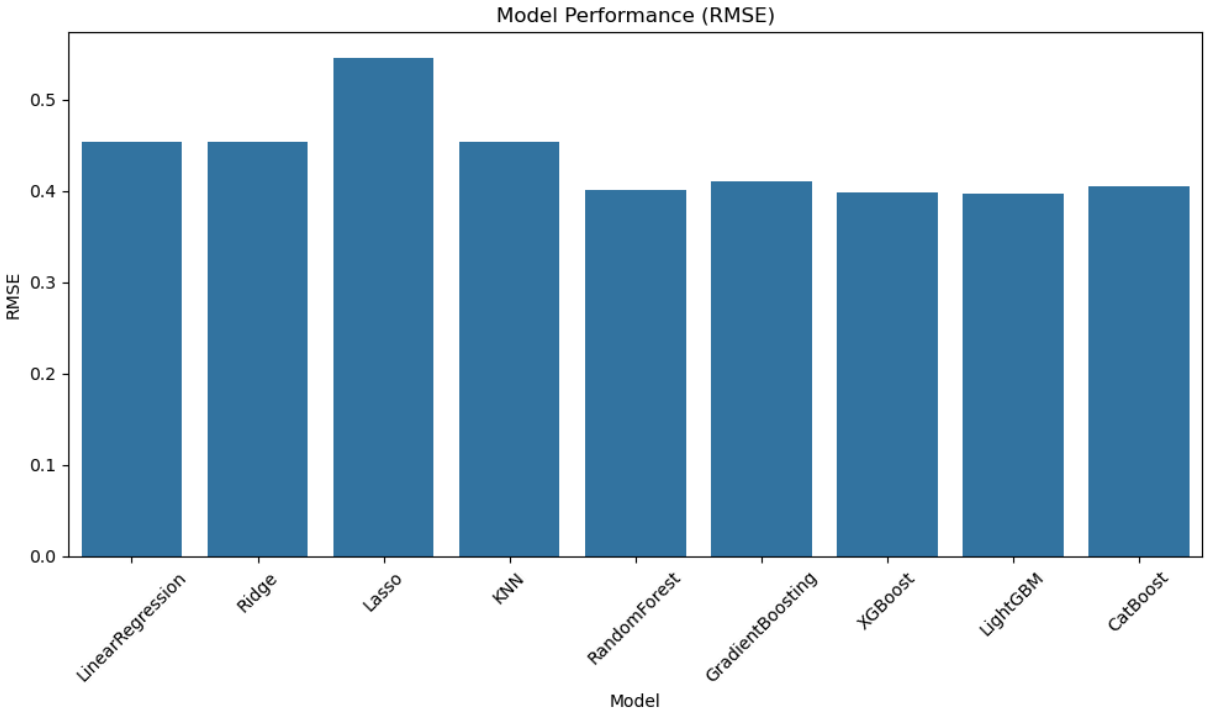
Model Performance Summary:

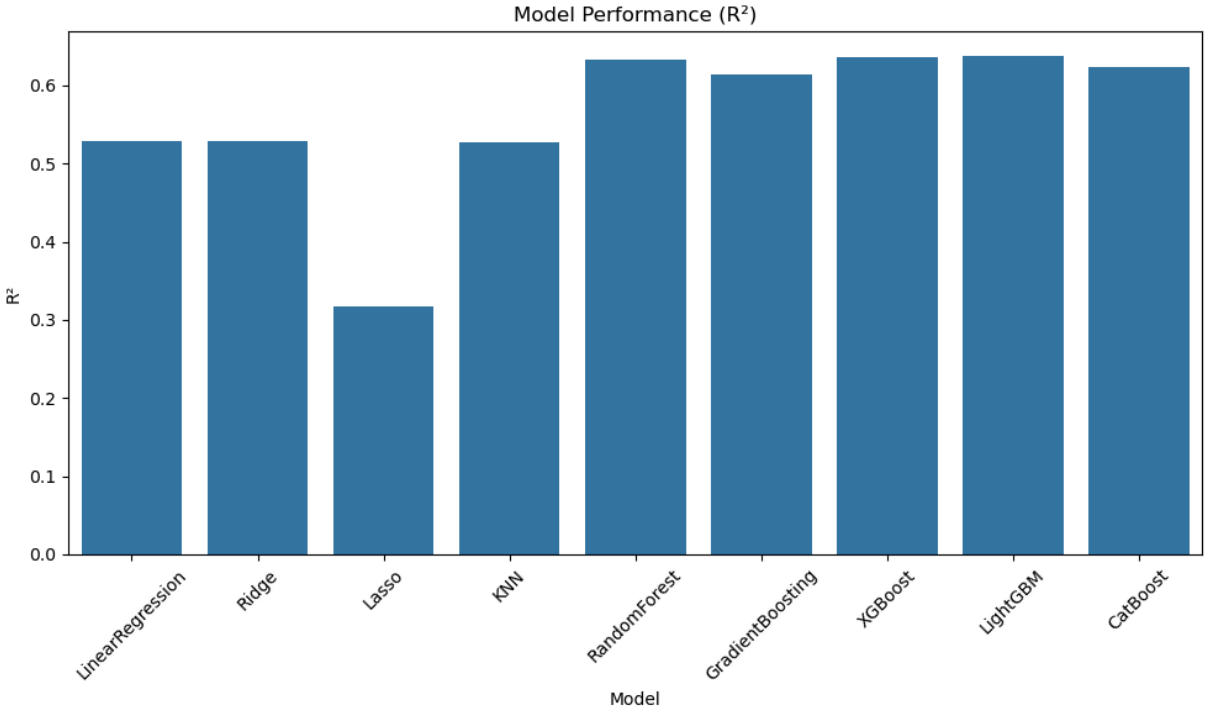
	Model	RMSE	R^2	MAE	MSE	MAPE (%)	CV_RMSE	Time
0	LinearRegression	0.4532	0.5292	0.3414	0.2054	7.2424	0.4599	0.2641
1	Ridge	0.4532	0.5293	0.3414	0.2054	7.2423	0.4599	0.5165
2	Lasso	0.5460	0.3167	0.4246	0.2981	9.1077	0.5501	0.4448
3	KNN	0.4538	0.5280	0.3458	0.2059	7.3408	0.4699	2.6352
4	RandomForest	0.4004	0.6325	0.2949	0.1603	6.2328	0.4275	158.0583
5	GradientBoosting	0.4104	0.6139	0.3055	0.1685	6.4425	0.4211	46.5996
6	XGBoost	0.3983	0.6364	0.2954	0.1586	6.2300	0.4141	2.4682
7	LightGBM	0.3975	0.6378	0.2954	0.1580	6.2336	0.4141	12.6407
8	CatBoost	0.4054	0.6234	0.3017	0.1643	6.3620	0.4177	8.0507

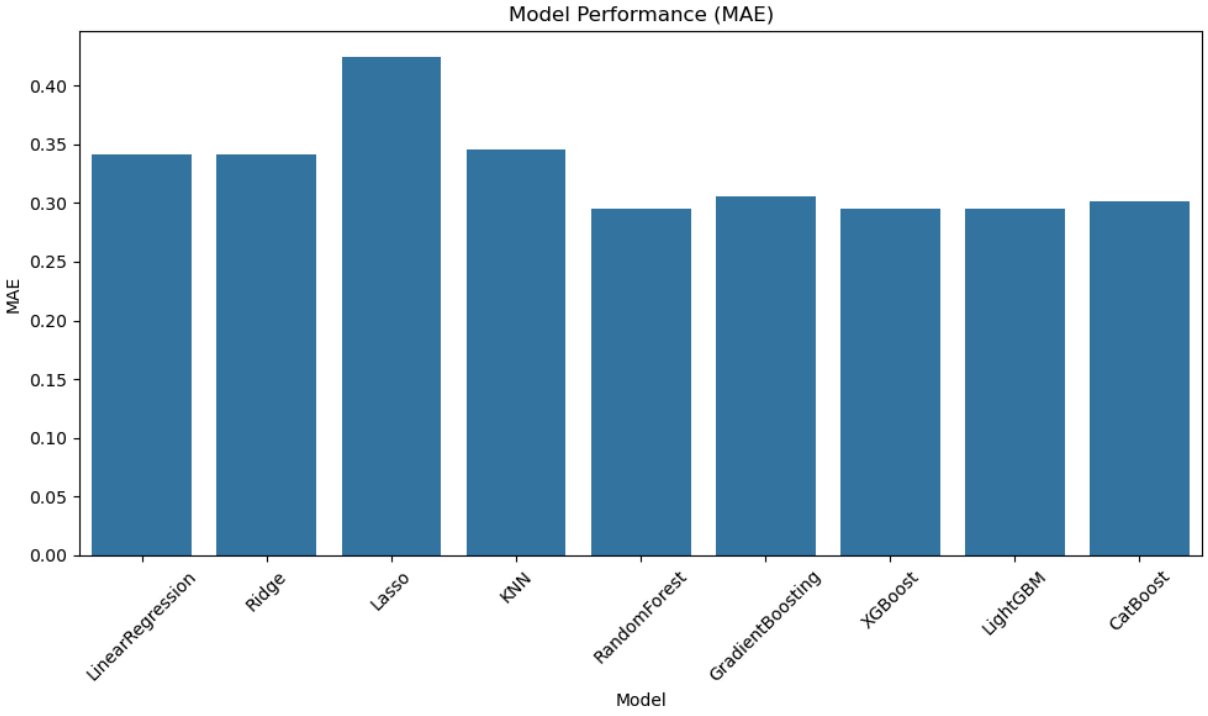
Model performance summary saved as 'model_performance_summary.csv'

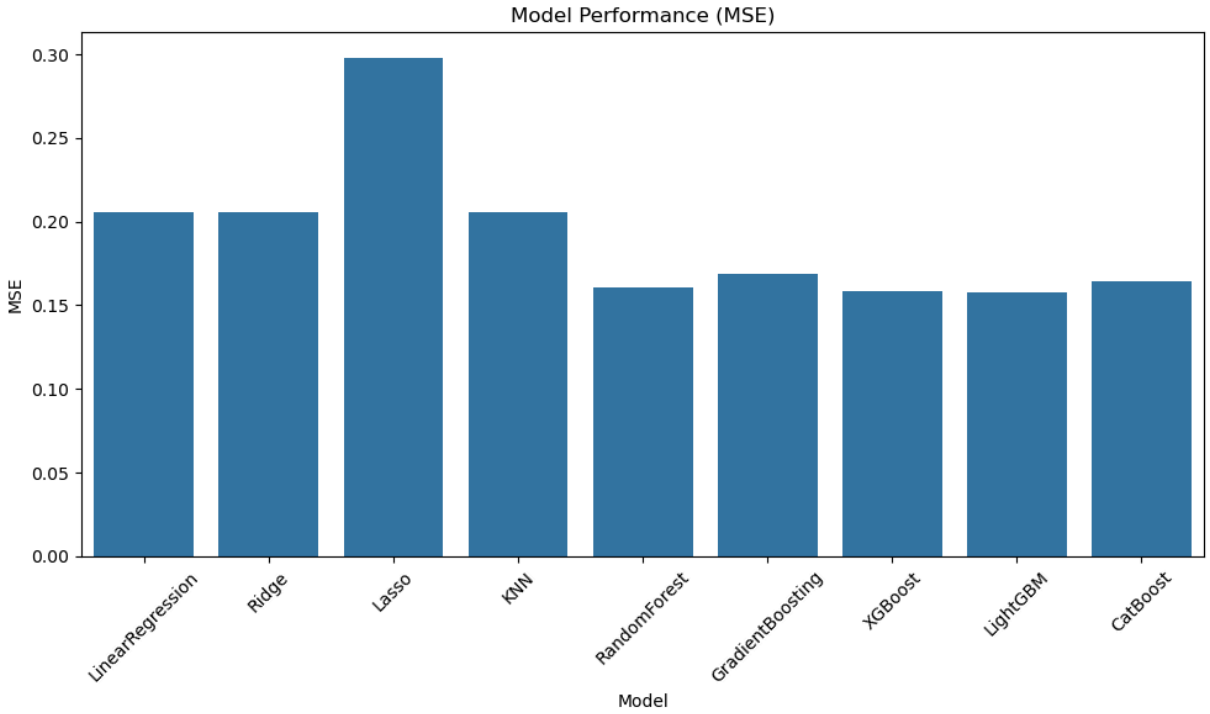
The model performance summary indicates that LightGBM achieved the best overall results among all tested models. It has the lowest RMSE (0.3975), lowest MSE (0.1580), and the highest R^2 score (0.6378), meaning it explains the largest proportion of variance in log_price. Additionally, its MAE (0.2954) and MAPE (6.23%) are among the lowest, suggesting it consistently makes accurate predictions. While its training time (14.14s) is longer than linear models, it's significantly faster than Gradient Boosting (47.22s), making LightGBM both effective and reasonably efficient. Given this balance of accuracy and speed, LightGBM is the most suitable model for predicting Airbnb listing prices.

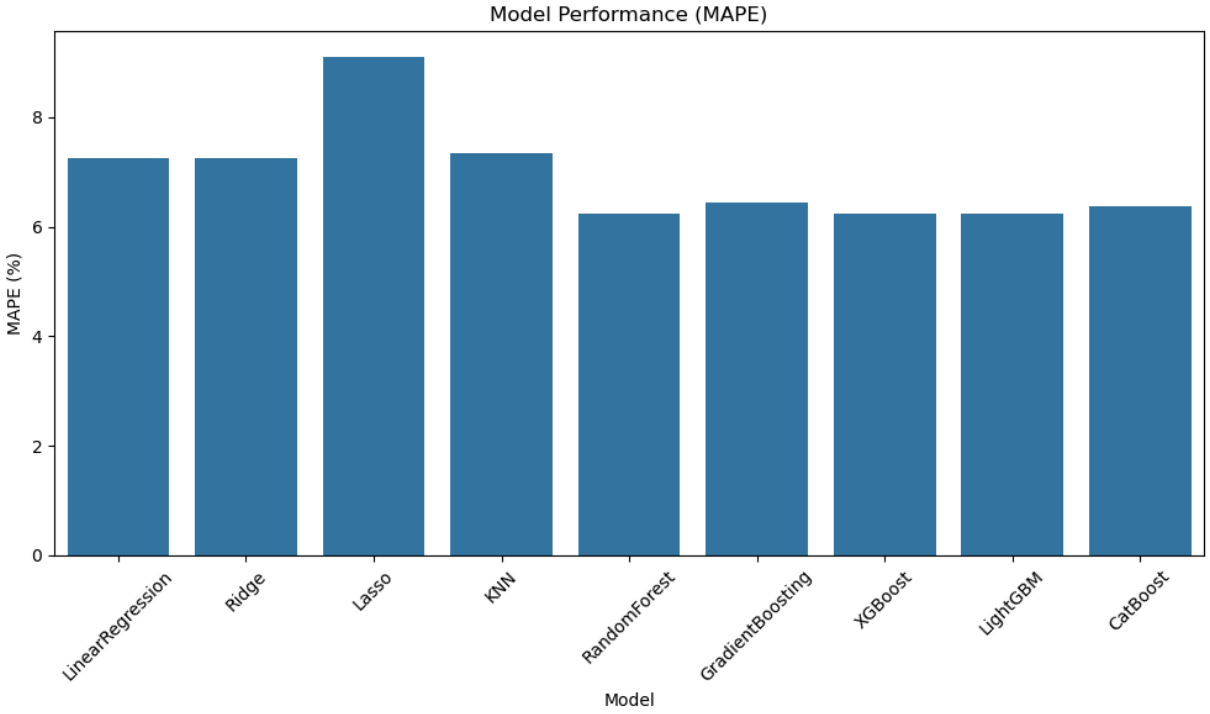
In [61]:
```
# Plotting Model Performance
def plot_metric(df, metric, title, ylabel):
    plt.figure(figsize=(10, 6))
    sns.barplot(x='Model', y=metric, data=df)
    plt.title(title)
    plt.ylabel(ylabel)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.savefig(f'figures/model_{metric.lower().replace(" (%)", "")}_compari
    plt.show()
    plt.close()

plot_metric(results_df, 'RMSE', 'Model Performance (RMSE)', 'RMSE')
plot_metric(results_df, 'R²', 'Model Performance (R²)', 'R²')
plot_metric(results_df, 'MAE', 'Model Performance (MAE)', 'MAE')
plot_metric(results_df, 'MSE', 'Model Performance (MSE)', 'MSE')
plot_metric(results_df, 'MAPE (%)', 'Model Performance (MAPE)', 'MAPE (%)')
plot_metric(results_df, 'CV_RMSE', 'Model Performance (CV RMSE)', 'CV RMSE')
plot_metric(results_df, 'Time', 'Model Training Time', 'Seconds')

# Detailed Analysis for Best Model
best_model_name = results_df.loc[results_df['RMSE'].idxmin(), 'Model']
print(f"\nBest Model: {best_model_name}")

# Train best model (using RandomForest as example for parameter extraction)
best_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(random_state=42, n_estimators=100))
])
best_pipeline.fit(X_train, y_train)
y_pred = best_pipeline.predict(X_test)
residuals = y_test - y_pred
```

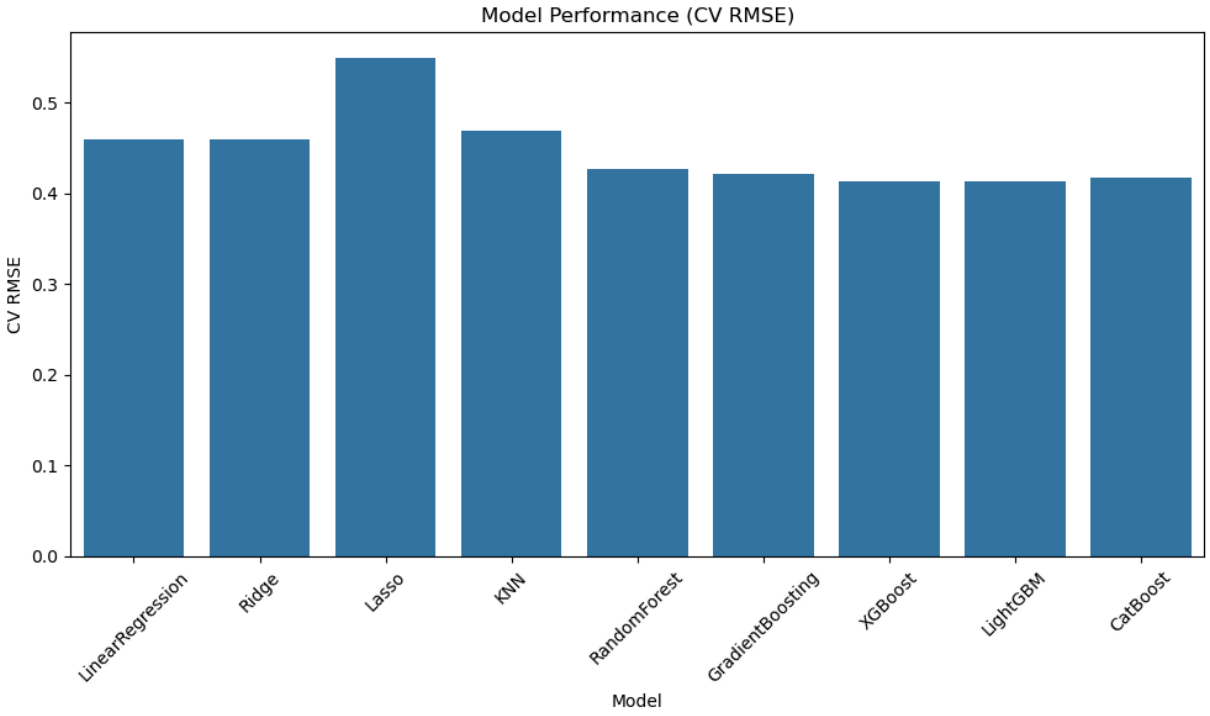


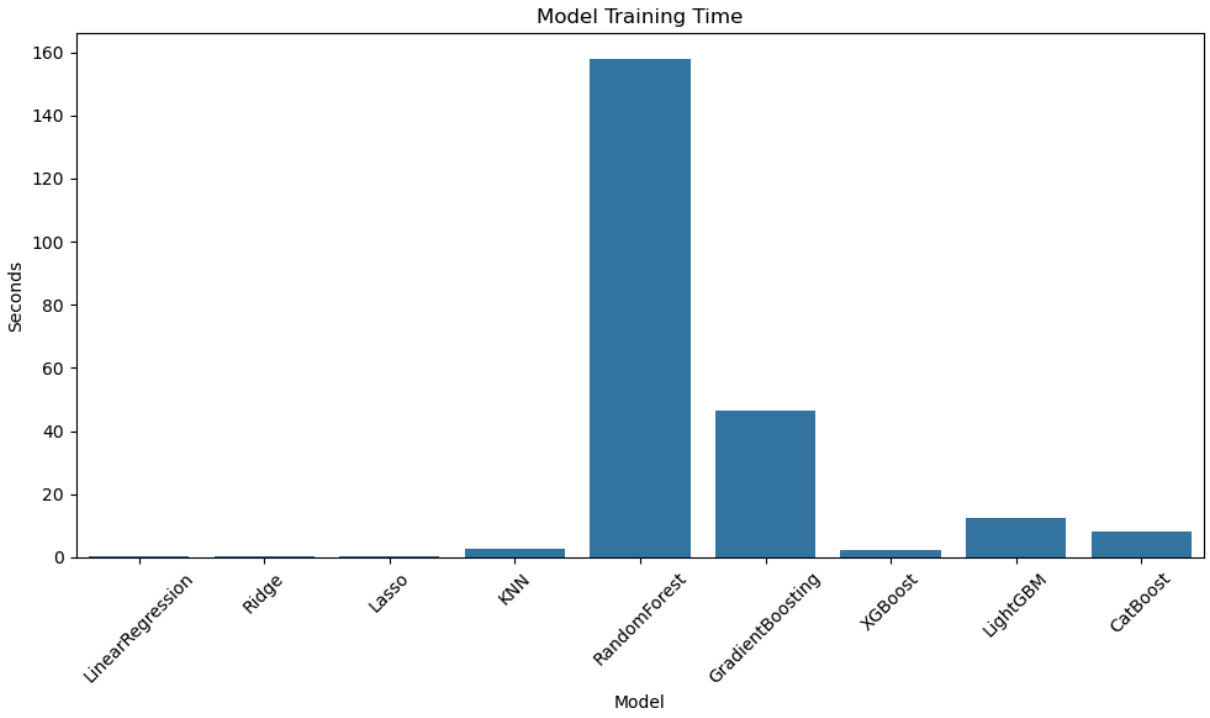

Model Performance (R²)
R²
0.0
0.1
0.2
0.3
0.4
0.5
0.6
LinearRegression
Ridge
Lasso
KNN
RandomForest
GradientBoosting
XGBoost
LightGBM
CatBoost
Model


Model Performance (MAE)
MAE
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
LinearRegression
Ridge
Lasso
KNN
RandomForest
GradientBoosting
XGBoost
LightGBM
CatBoost
Model


Model Performance (MSE)
MSE
0.30
0.25
0.20
0.15
0.10
0.05
0.00
LinearRegression
Ridge
Lasso
KNN
RandomForest
GradientBoosting
XGBoost
LightGBM
CatBoost
Model

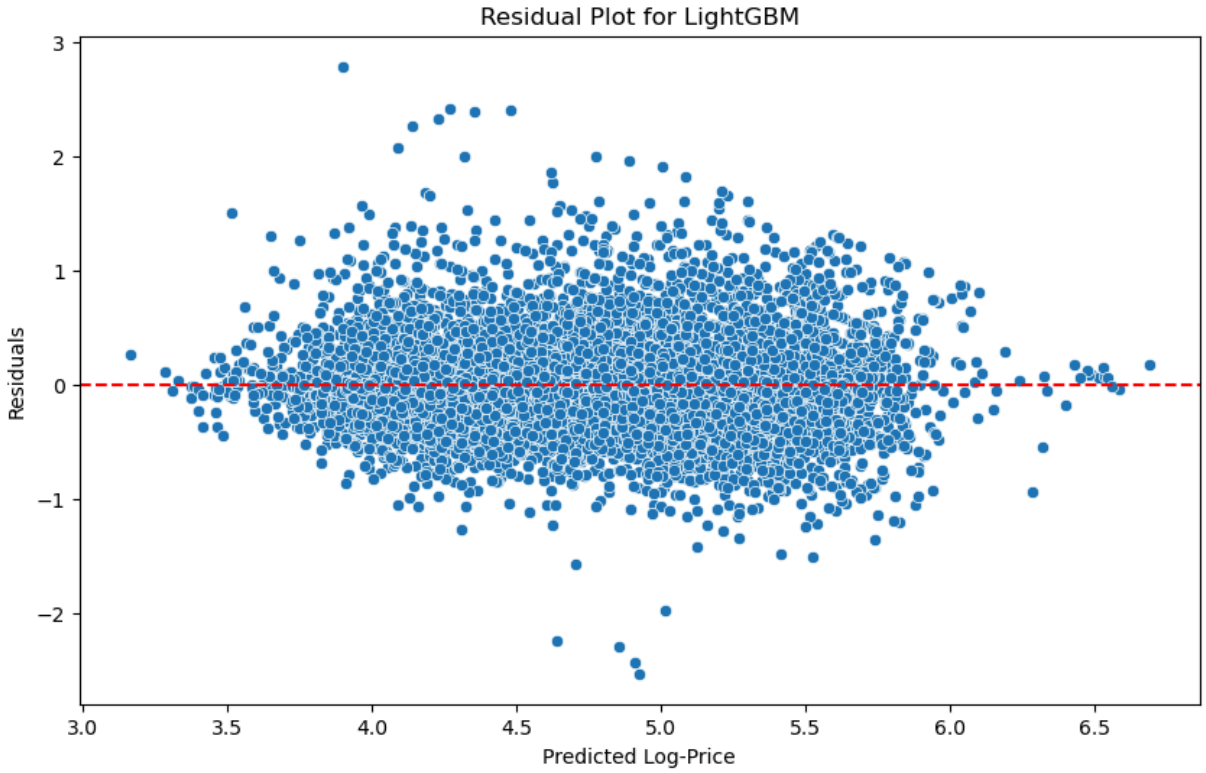

Model Performance (MAPE)
MAPE (%)
8
6
4
2
0
LinearRegression
Ridge
Lasso
KNN
RandomForest
GradientBoosting
XGBoost
LightGBM
CatBoost
Model



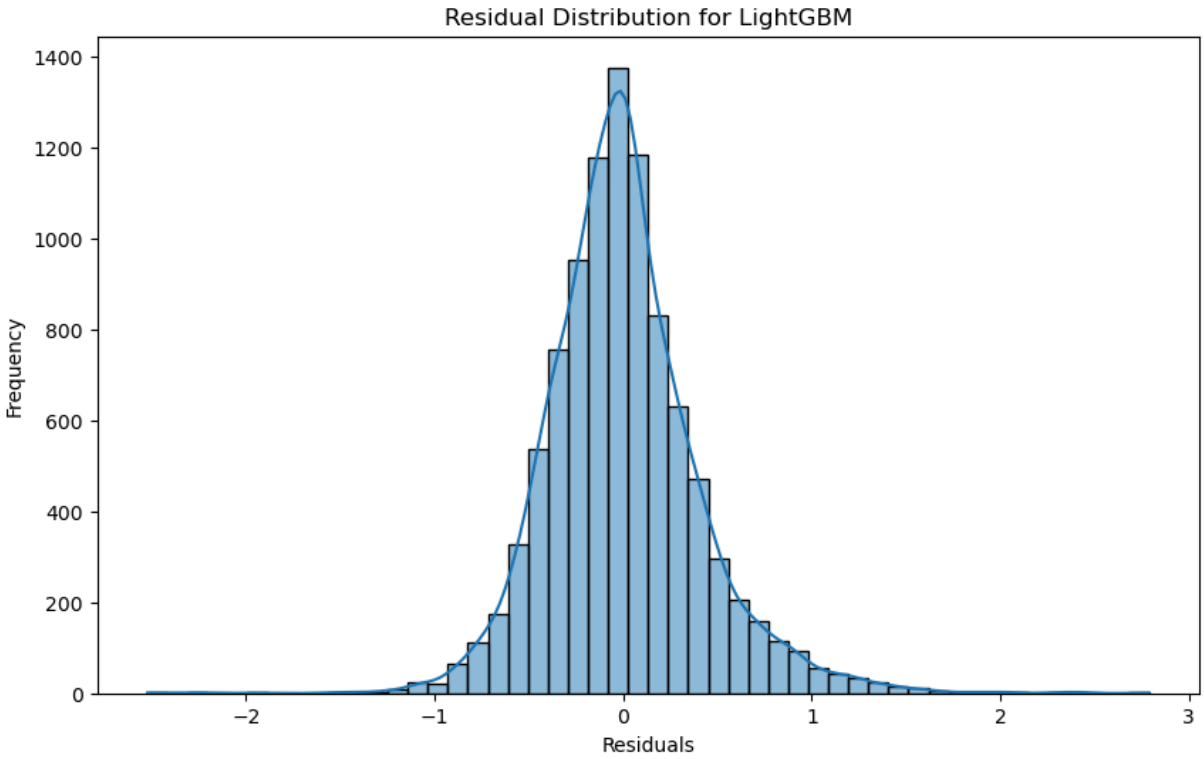



```
Best Model: LightGBM
```

In [62]:
```
# Residual Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_pred, y=residuals)
plt.axhline(0, color='red', linestyle='--')
plt.title(f'Residual Plot for {best_model_name}')
plt.xlabel('Predicted Log-Price')
plt.ylabel('Residuals')
plt.savefig('figures/residual_plot.png')
plt.show()
plt.close()
```

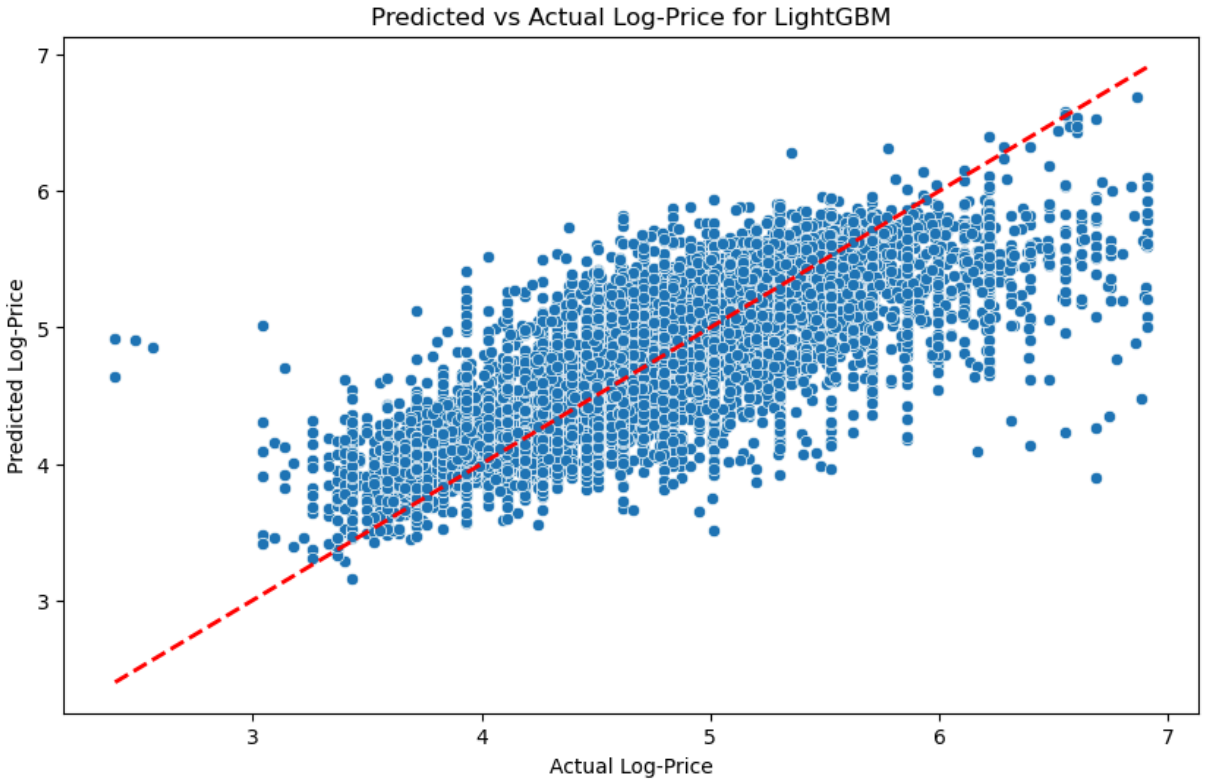


In [63]:
```
# Residual Histogram
plt.figure(figsize=(10, 6))
sns.histplot(residuals, bins=50, kde=True)
plt.title(f'Residual Distribution for {best_model_name}')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.savefig('figures/residual_histogram.png')
plt.show()
plt.close()
```



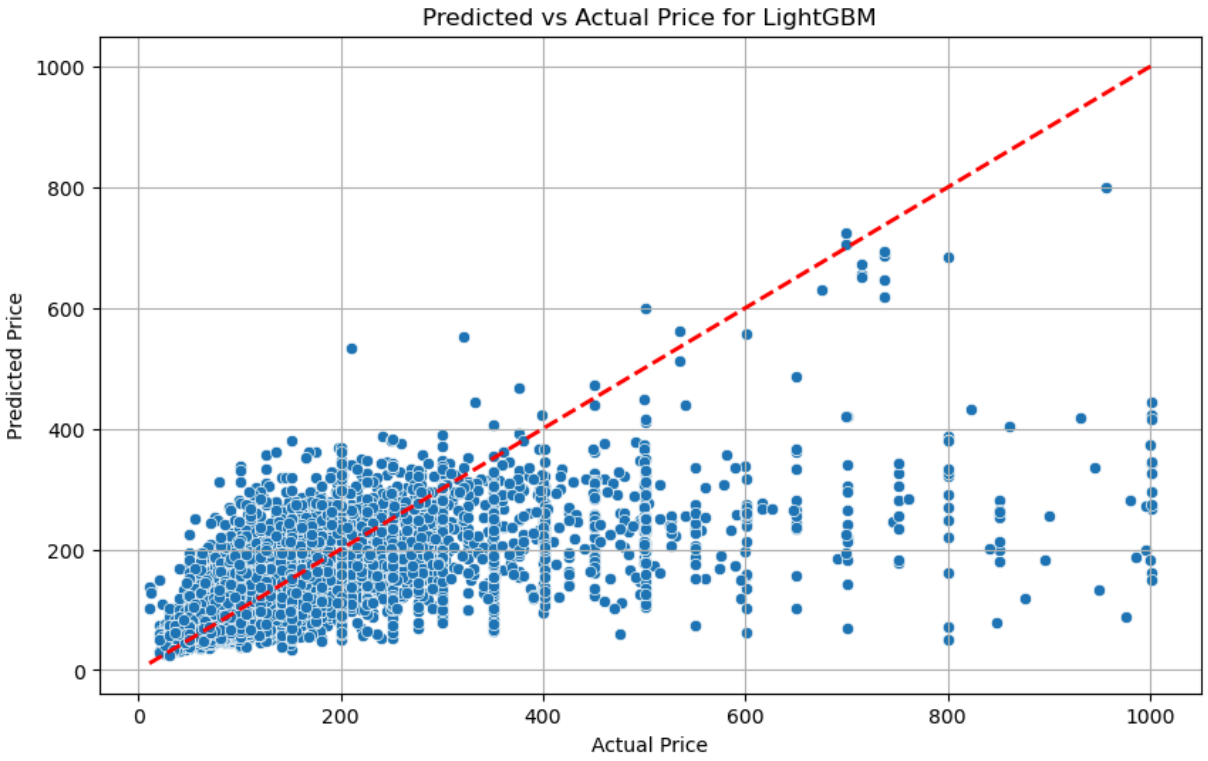
The residual plots indicate that the LightGBM model performs well in predicting log-transformed prices. In the residuals vs. predicted values plot, the residuals are evenly scattered around zero with no clear pattern, suggesting the model does not suffer from systematic bias and maintains consistent error variance across the prediction range. The histogram of residuals further supports this, showing a roughly symmetric, bell-shaped distribution centered near zero—indicating that the errors are approximately normally distributed. Together, these plots suggest that the model's predictions are reliable, with errors that are random and well-behaved.

In [64]:
```
# Predicted vs Actual Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--',
plt.title(f'Predicted vs Actual Log-Price for {best_model_name}')
plt.xlabel('Actual Log-Price')
plt.ylabel('Predicted Log-Price')
plt.savefig('figures/predicted_vs_actual.png')
plt.show()
plt.close()
```



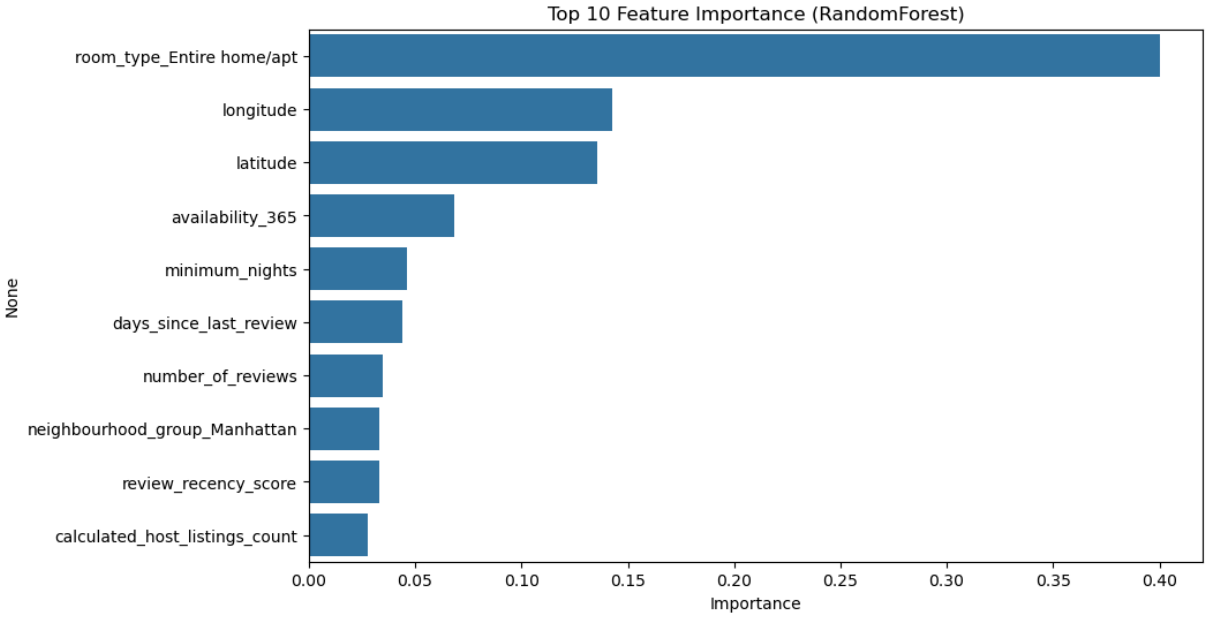
In [67]:
```
# Convert log_price back to price for better interpretability
y_test_actual_price = np.exp(y_test)
y_pred_actual_price = np.exp(y_pred)

plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test_actual_price, y=y_pred_actual_price)
plt.plot([y_test_actual_price.min(), y_test_actual_price.max()],
         [y_test_actual_price.min(), y_test_actual_price.max()],
         'r--', lw=2)
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title(f'Predicted vs Actual Price for {best_model_name}')
plt.grid(True)
plt.savefig('figures/predicted_vs_actual_price.png')
plt.show()


# Create a DataFrame comparing actual vs predicted prices
comparison_df = pd.DataFrame({
    "Actual Price": y_test_actual_price,
    "Predicted Price": y_pred_actual_price
})
comparison_df["Absolute Error"] = np.abs(comparison_df["Actual Price"] - com
comparison_df = comparison_df.reset_index(drop=True)
print("First 10 Predictions vs Actuals:")
print(comparison_df.head(10))

# Calculate error metrics on actual prices
mae = mean_absolute_error(y_test_actual_price, y_pred_actual_price)
mse = mean_squared_error(y_test_actual_price, y_pred_actual_price)
rmse = np.sqrt(mse)
```

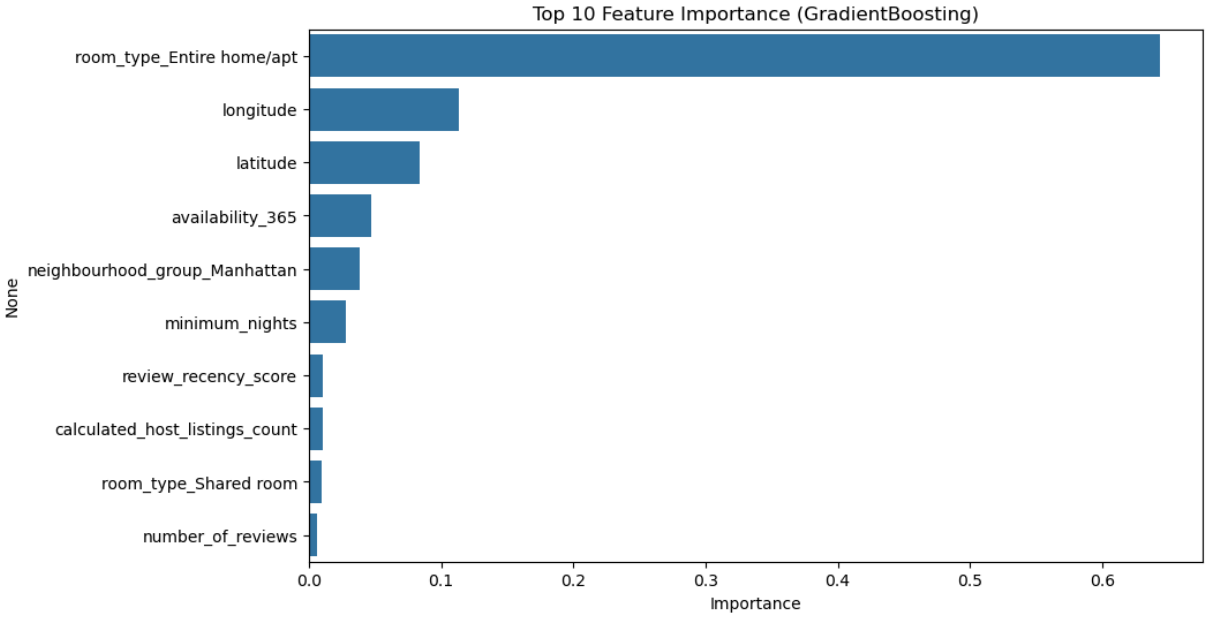
```
r2 = r2_score(y_test_actual_price, y_pred_actual_price)
mape = np.mean(np.abs((y_test_actual_price - y_pred_actual_price) / y_test_a

print("\nEvaluation on Actual Price:")
print(f"MAE  (Mean Absolute Error):      ${mae:.2f}")
print(f"R²   (R-squared Score):          {r2:.4f}")
print(f"MAPE (Mean Absolute % Error):    {mape:.2f}%")

# Create a DataFrame with evaluation results
metrics_df = pd.DataFrame({
    "Metric": ["MAE", "MSE", "RMSE", "R²", "MAPE"],
    "Value": [mae, mse, rmse, r2, mape]
})
metrics_df.to_csv("best_price_evaluation.csv", index=False)
```

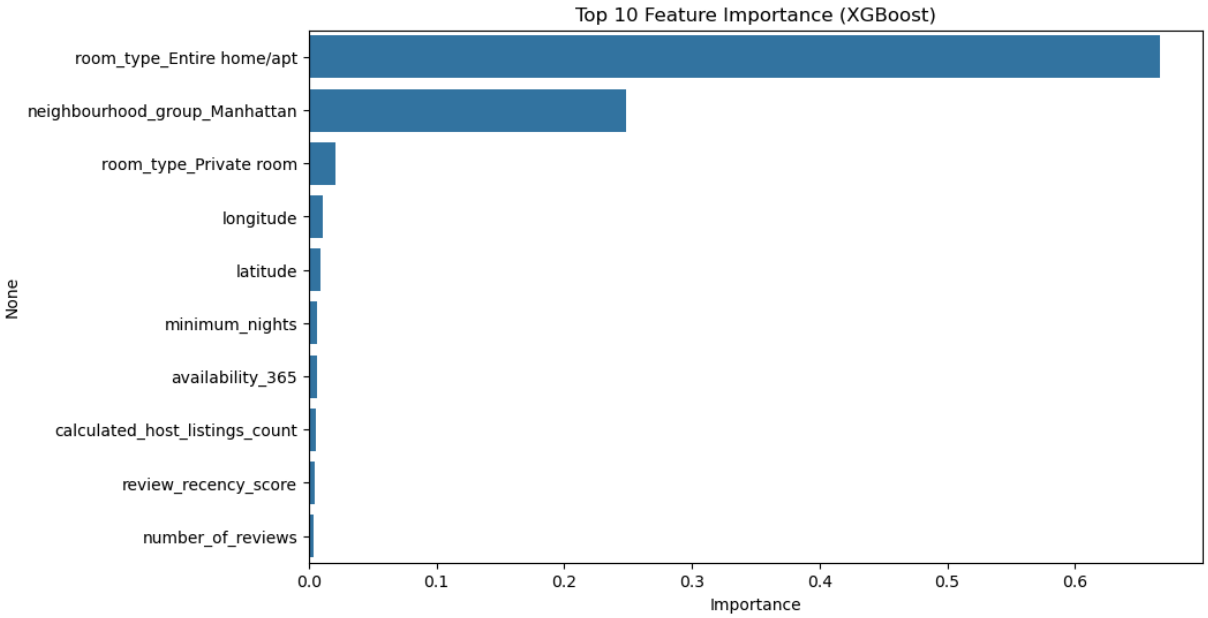



```
First 10 Predictions vs Actuals:
   Actual Price  Predicted Price  Absolute Error
0          56.0        63.008207        7.008207
1         101.0        77.658035       23.341965
2         166.0       114.987912       51.012088
3          61.0        55.816562        5.183438
4         191.0       224.252822       33.252822
5          51.0        69.427480       18.427480
6          76.0       107.150011       31.150011
7          46.0        52.272716        6.272716
8         251.0       222.484075       28.515925
9         296.0       276.255507       19.744493

Evaluation on Actual Price:
MAE  (Mean Absolute Error):      $45.50
R²   (R-squared Score):          0.4574
MAPE (Mean Absolute % Error):    30.49%
```

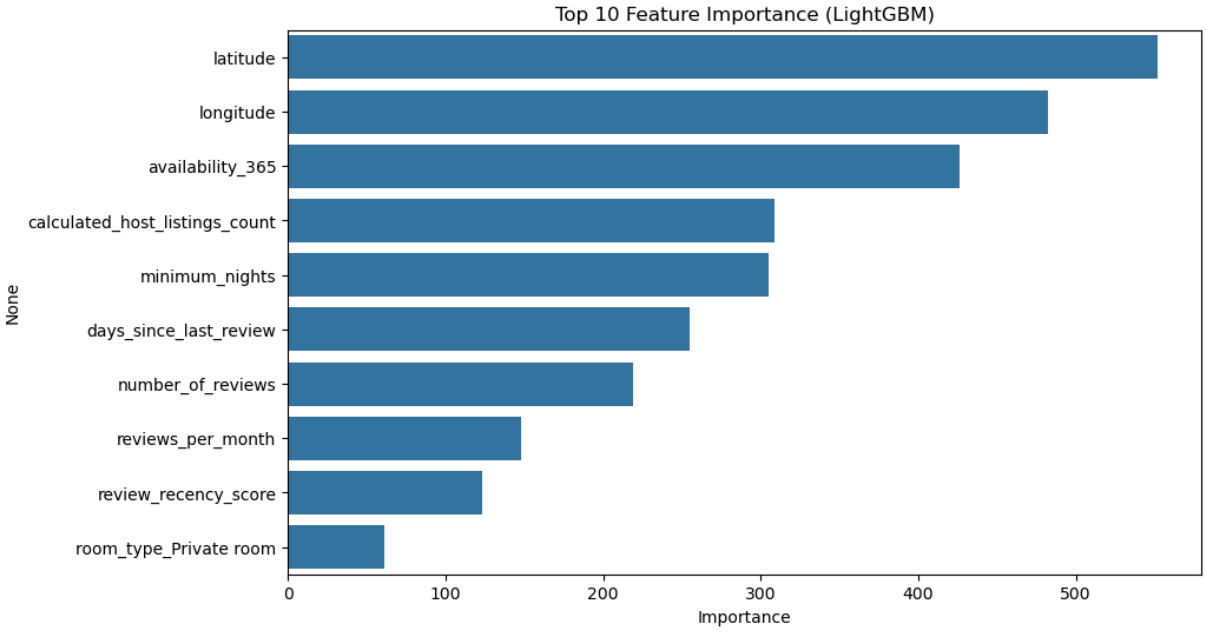
In [66]:
```
# Feature Importance for Tree-Based Models
tree_based_models = ['RandomForest', 'GradientBoosting', 'XGBoost', 'LightGB
feature_names = (numeric_features +
                 best_pipeline.named_steps['preprocessor'].named_transformer
                 passthrough_features)

for name, model in models:
    if name in tree_based_models:
        pipeline = Pipeline([
            ('preprocessor', preprocessor),
            ('regressor', model)
        ])
        pipeline.fit(X_train, y_train)
        importances = pipeline.named_steps['regressor'].feature_importances_
        feature_importance = pd.Series(importances, index=feature_names).sor

        plt.figure(figsize=(10, 6))
        sns.barplot(x=feature_importance.values, y=feature_importance.index)
        plt.title(f'Top 10 Feature Importance ({name})')
        plt.xlabel('Importance')
        plt.savefig(f'figures/feature_importance_{name.lower()}.png')
        plt.show()
        plt.close()
```

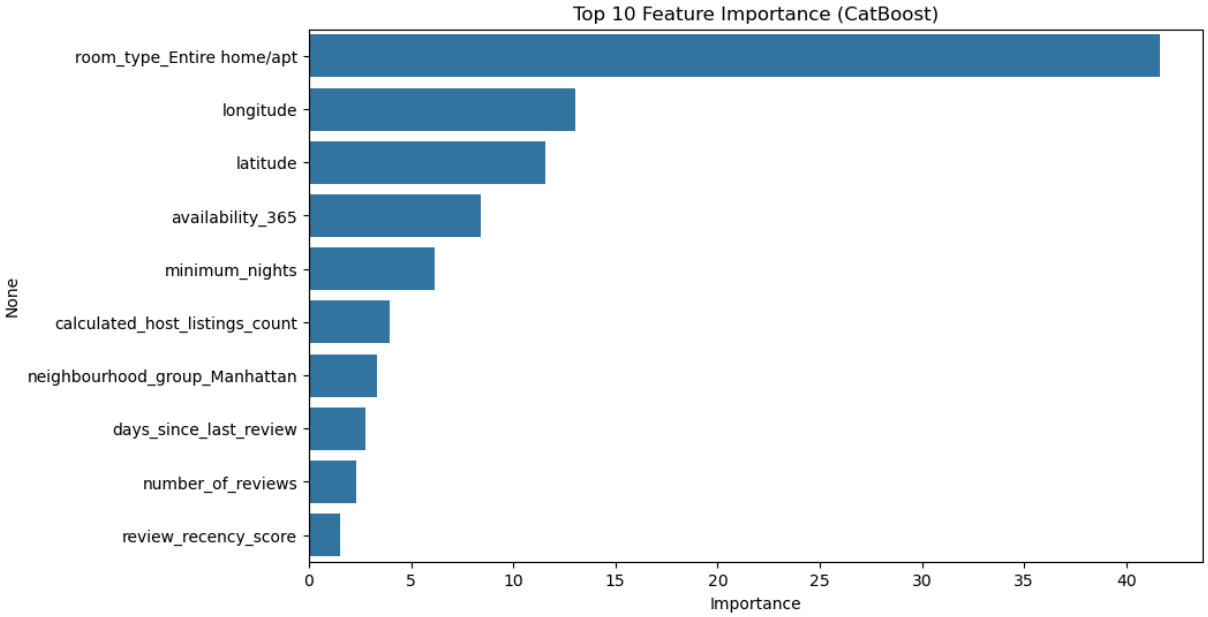







```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000831 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1910
[LightGBM] [Info] Number of data points in the train set: 38892, number of u
sed features: 18
[LightGBM] [Info] Start training from score 4.722582
```





The feature importance plot for LightGBM, our selected best-performing model, highlights the most influential variables in predicting Airbnb log prices. Unlike other models that place dominant weight on categorical features like room_type_Entire home/apt, LightGBM assigns the highest importance to geolocation features — specifically latitude and longitude. This suggests that spatial location plays the most critical role in determining listing prices, which aligns with real-world expectations where central or desirable areas command higher prices.

Other important features include availability_365, calculated_host_listings_count, and minimum_nights, reflecting how booking frequency, host scale, and stay restrictions influence pricing. Temporal review features like days_since_last_review and review_recency_score also appear among the top 10, indicating some impact from listing activity and freshness. Interestingly, categorical room types (e.g., room_type_Private

room) have comparatively low influence in LightGBM, in contrast to tree-based models like RandomForest or GradientBoosting.

Overall, LightGBM reveals a more nuanced balance across spatial, behavioral, and review-related features, reinforcing its ability to capture complex pricing patterns beyond just room category.