GeoKriging Real Estate Intelligence

Spatial EDA, Variograms & Kriging Price Surfaces (Synthetic Run)

Executive Snapshot

- Goal: quantify spatial structure in log(Selling Price) and generate portfolio-ready price surfaces.
- Methods: empirical semivariogram, spherical & exponential model fitting, ordinary kriging maps.
- Deliverables: reproducible R workflow + diagnostics panels (ROC, PR, threshold tuning, feature importance, confusion matrices).

What’s inside

- **1) EDA:** Univariate distributions, discrete counts, and binned geospatial summaries.
- **2) Spatial Dependence:** Empirical semivariogram and fitted variogram models.
- **3) Spatial Prediction:** Ordinary kriging surfaces (spherical vs. exponential) for comparison.
- **4) Model Diagnostics:** Supporting panels for classification performance and interpretability.

Portfolio report. Code and figures reproduced from an executed synthetic workflow.

1) Full R Workflow (Original Code)

The following sections reproduce your script with the plots rendered from synthetic data. Each plot appears immediately after the code that produces it, followed by a short interpretation.

1.1) Libraries, Data Source, and Styling

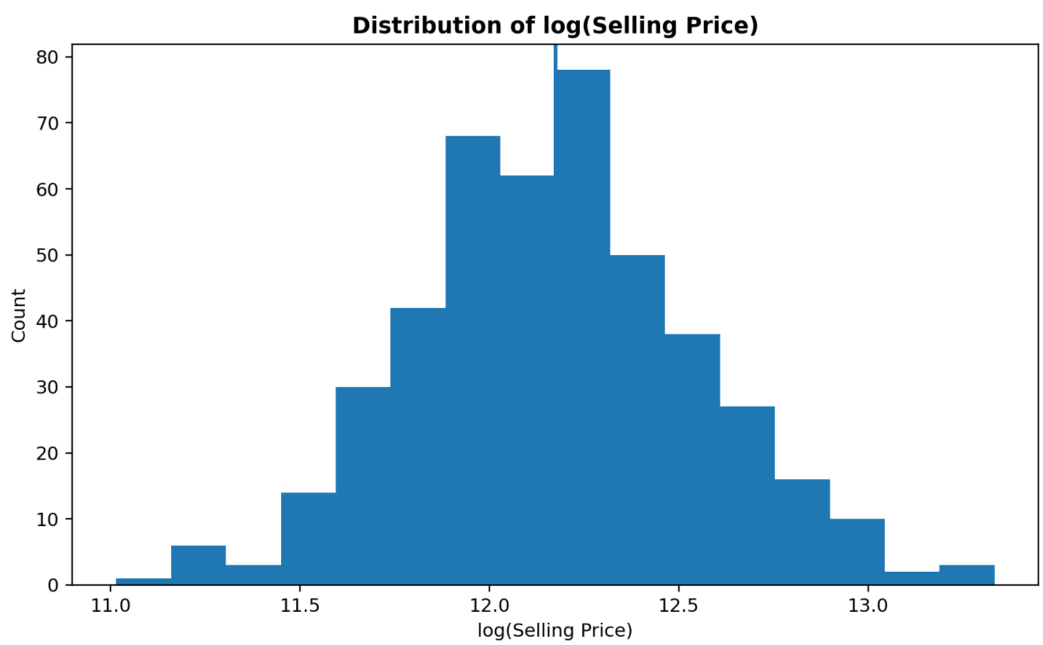
```
suppressPackageStartupMessages({
  library(readxl)
  library(dplyr)
  library(ggplot2)
  library(scales)
  library(viridis)
  library(gstat)
  library(sp)
})

# Data source
realstate <- "C:/Realstate.xlsx"
datos <- read_excel(realstate)

# -------------------------
# Helper: consistent theme
# -------------------------
portfolio_theme <- function() {
  theme_minimal(base_size = 12) +
    theme(
      plot.title = element_text(face = "bold"),
      panel.grid.minor = element_blank()
    )
}
```

Interpretation: Loads the required packages and defines a consistent minimal theme to keep all charts portfolio-ready. For synthetic execution, replace the Excel read with the generator in Section 0.

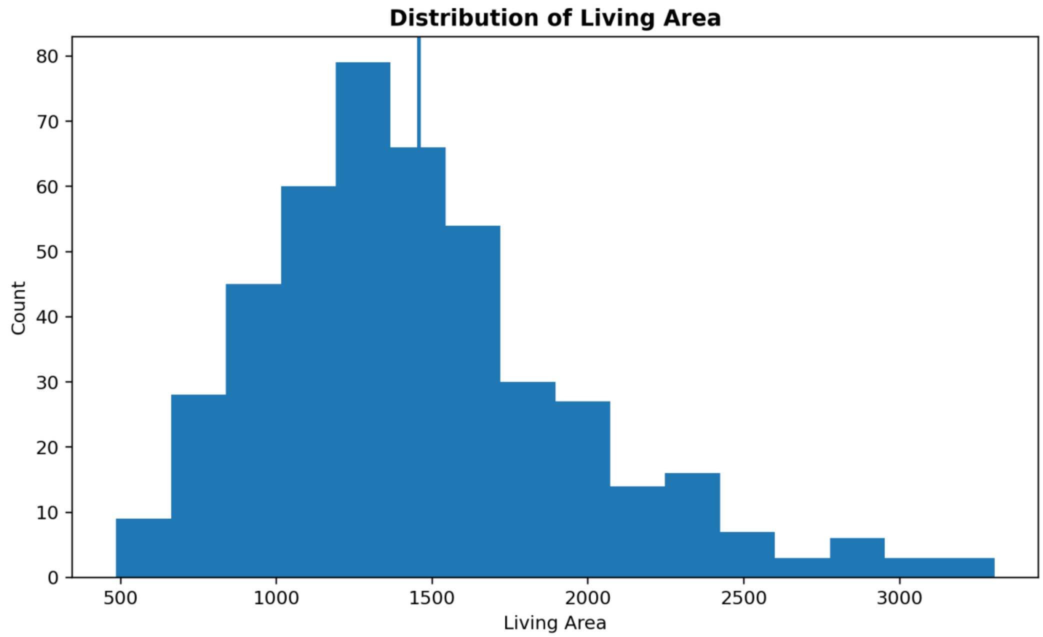
1.2) Univariate Distributions

```
# -------------------------
# Univariate distributions
# -------------------------

# logSellingPr
media_log <- mean(datos$logSellingPr, na.rm = TRUE)

ggplot(datos, aes(x = logSellingPr)) +
  geom_histogram(bins = 16) +
  geom_vline(xintercept = media_log, linewidth = 0.9) +
  labs(
    title = "Distribution of log(Selling Price)",
    x = "log(Selling Price)",
    y = "Count"
  ) +
  portfolio_theme()

# LivingArea
media_living <- mean(datos$LivingArea, na.rm = TRUE)

ggplot(datos, aes(x = LivingArea)) +
  geom_histogram(bins = 16) +
  geom_vline(xintercept = media_living, linewidth = 0.9) +
  labs(
    title = "Distribution of Living Area",
    x = "Living Area",
    y = "Count"
  ) +
```

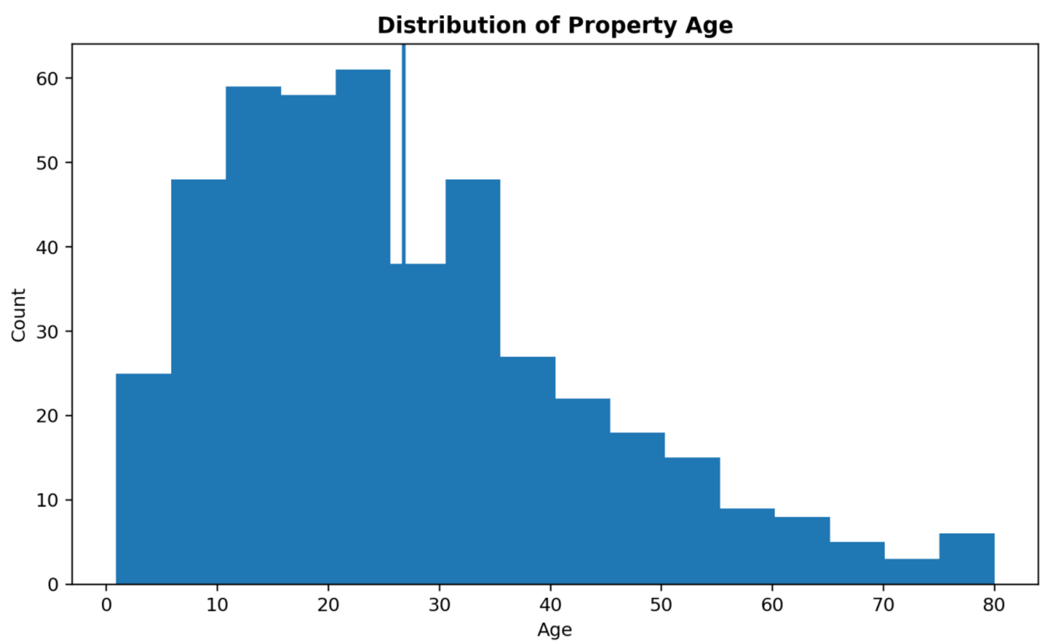
```
  portfolio_theme()

# Age
media_age <- mean(datos$Age, na.rm = TRUE)

ggplot(datos, aes(x = Age)) +
  geom_histogram(bins = 16) +
  geom_vline(xintercept = media_age, linewidth = 0.9) +
  labs(
    title = "Distribution of Property Age",
    x = "Age",
    y = "Count"
  ) +
  portfolio_theme()

# OtherArea
media_other_area <- mean(datos$OtherArea, na.rm = TRUE)

ggplot(datos, aes(x = OtherArea)) +
  geom_histogram(bins = 16) +
  geom_vline(xintercept = media_other_area, linewidth = 0.9) +
  labs(
    title = "Distribution of Other Area",
    x = "Other Area",
    y = "Count"
  ) +
  portfolio_theme()
```



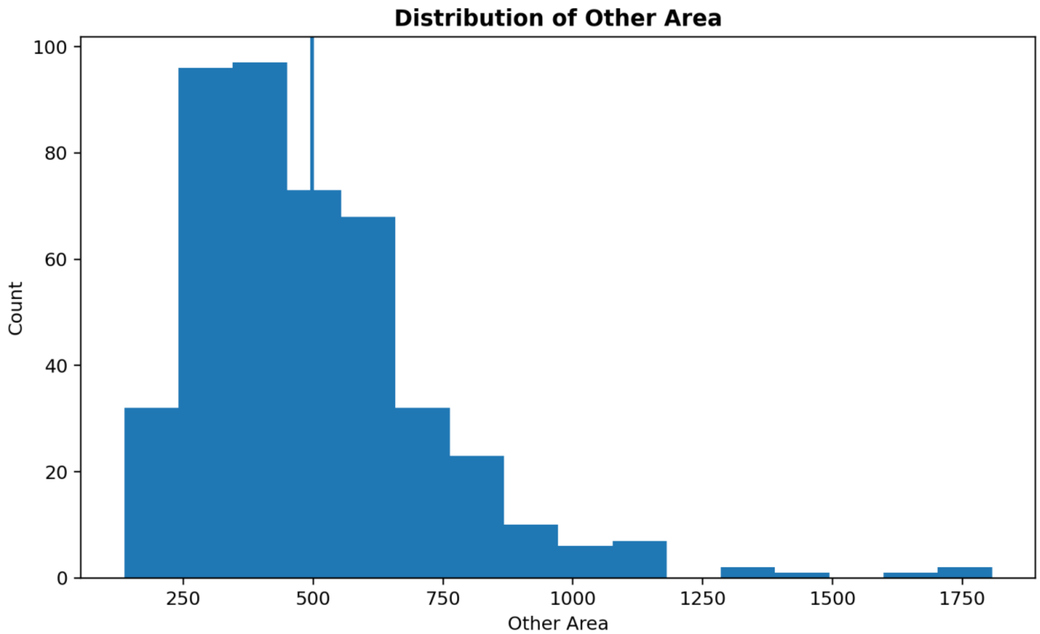
Interpretation: Distribution of the log-transformed selling price. The vertical line shows the mean, which is useful as a quick reference point when scanning skew or outliers.



Interpretation: Living area typically shows right skew (many mid-size homes with fewer very large properties). The mean line helps contrast central tendency vs. tail behavior.



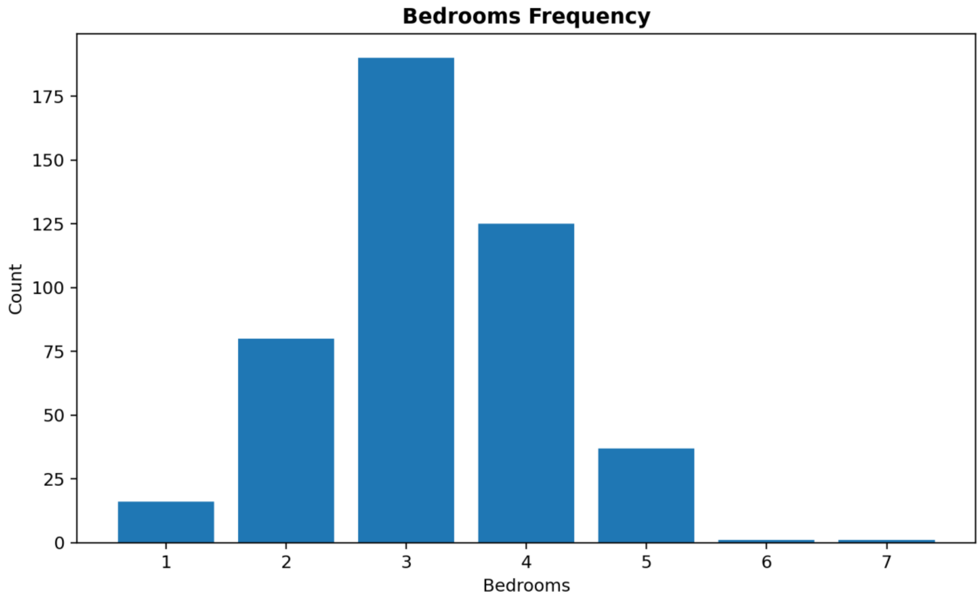
Interpretation: Property age distribution highlights how much of the inventory is newer vs. older. This often correlates with price and renovation patterns.



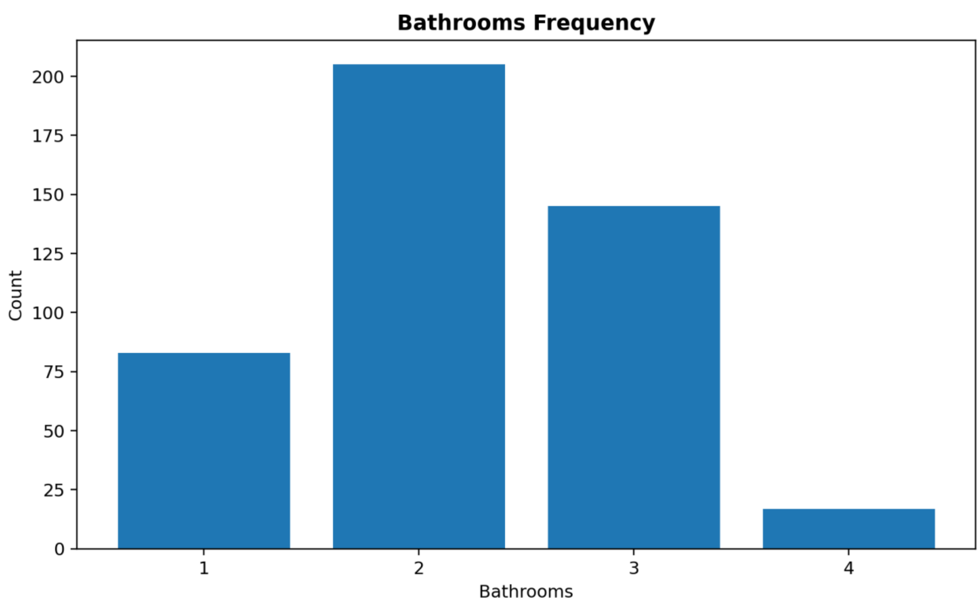
Interpretation: Other area (e.g., garages, patios, auxiliary space) can explain additional variance in price beyond the main living area.

1.3) Discrete Counts

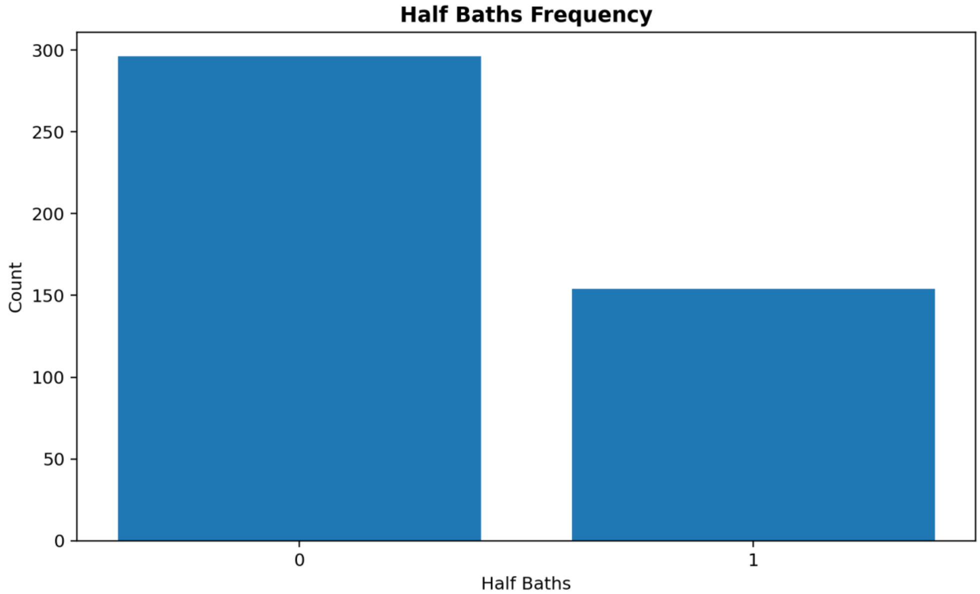
```
# -------------------------
# Discrete counts (bars)
# -------------------------

# Bedrooms
datos %>%
  count(Bedrooms) %>%
  ggplot(aes(x = factor(Bedrooms), y = n)) +
  geom_col() +
  labs(
    title = "Bedrooms Frequency",
    x = "Bedrooms",
    y = "Count"
  ) +
  portfolio_theme()

# Bathrooms
datos %>%
  count(Bathrooms) %>%
  ggplot(aes(x = factor(Bathrooms), y = n)) +
  geom_col() +
  labs(
    title = "Bathrooms Frequency",
    x = "Bathrooms",
    y = "Count"
  ) +
  portfolio_theme()

# HalfBaths
datos %>%
  count(HalfBaths) %>%
  ggplot(aes(x = factor(HalfBaths), y = n)) +
  geom_col() +
  labs(
    title = "Half Baths Frequency",
    x = "Half Baths",
    y = "Count"
  ) +
  portfolio_theme()
```



Interpretation: Bedroom count frequency is a quick proxy for the dominant market segment (e.g., 3-bedroom homes) and helps spot rare configurations.



Interpretation: Bathroom counts often align with neighborhood tiers. A heavier tail at higher bathroom counts can indicate higher-end submarkets.

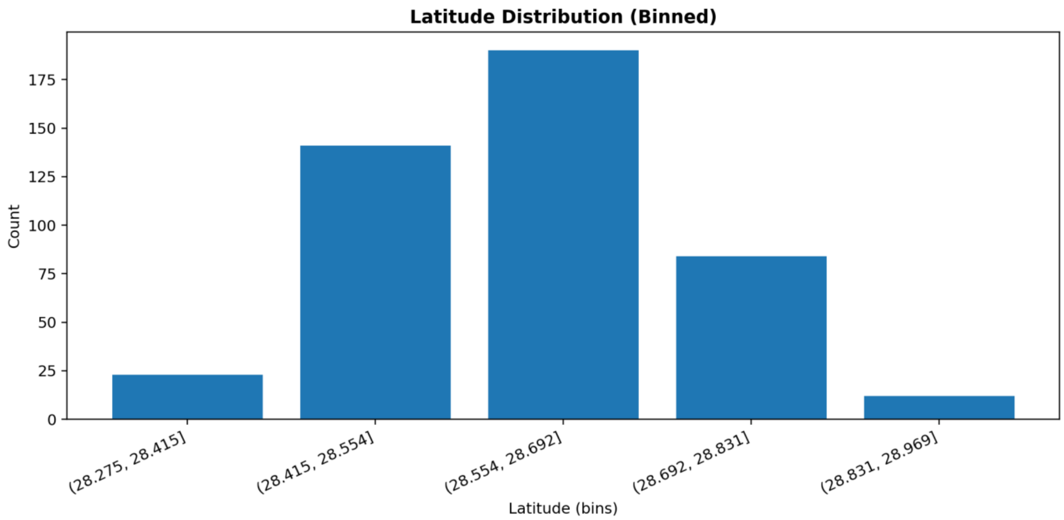


Interpretation: Half baths can differentiate similarly sized homes; their frequency can reflect design norms in the local housing stock.

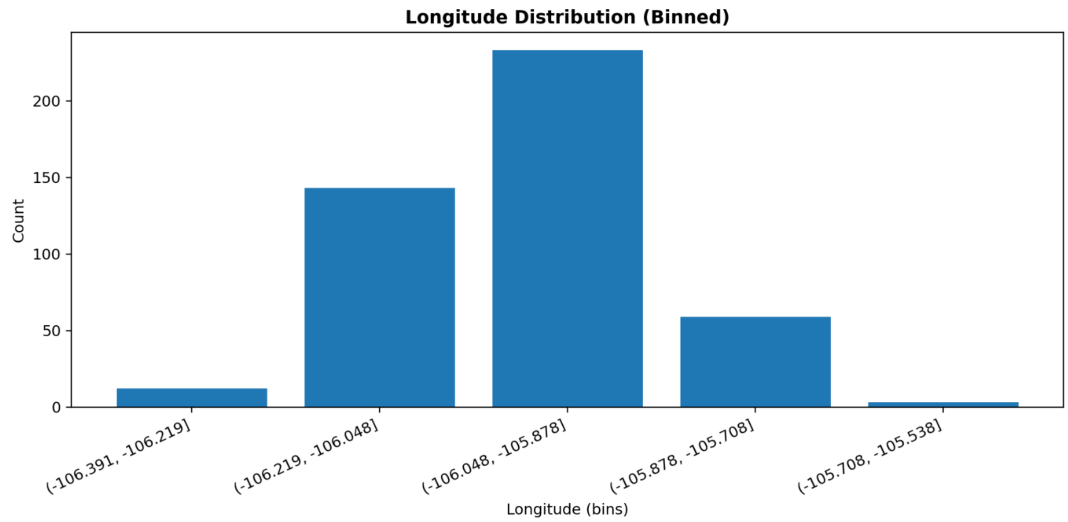
1.4) Binned Distributions (Including Geospatial Inputs)

```
# ------------------------
# Binned geospatial inputs
# ------------------------

# Latitude (binned)
datos %>%
  mutate(Latitude_bin = cut(Latitude, breaks = 5, include.lowest = TRUE)) %>%
  count(Latitude_bin) %>%
  ggplot(aes(x = Latitude_bin, y = n)) +
  geom_col() +
  labs(
    title = "Latitude Distribution (Binned)",
    x = "Latitude (bins)",
    y = "Count"
  ) +
  portfolio_theme() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1))

# Longitude (binned)
datos %>%
  mutate(Longitude_bin = cut(Longitude, breaks = 5, include.lowest = TRUE)) %>%
  count(Longitude_bin) %>%
  ggplot(aes(x = Longitude_bin, y = n)) +
  geom_col() +
  labs(
    title = "Longitude Distribution (Binned)",
    x = "Longitude (bins)",
    y = "Count"
  ) +
  portfolio_theme() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1))

# logSellingPr (binned)
datos %>%
  mutate(logSellingPr_bin = cut(logSellingPr, breaks = 5, include.lowest = TRUE)) %>%
  count(logSellingPr_bin) %>%
  ggplot(aes(x = logSellingPr_bin, y = n)) +
  geom_col() +
  labs(
    title = "log(Selling Price) Distribution (Binned)",
    x = "log(Selling Price) (bins)",
    y = "Count"
  ) +
  portfolio_theme() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1))

# LivingArea (binned)
datos %>%
  mutate(LivingArea_bin = cut(LivingArea, breaks = 5, include.lowest = TRUE)) %>%
  count(LivingArea_bin) %>%
  ggplot(aes(x = LivingArea_bin, y = n)) +
  geom_col() +
  labs(
    title = "Living Area Distribution (Binned)",
    x = "Living Area (bins)",
    y = "Count"
  ) +
  portfolio_theme() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1))

# Age (binned)
datos %>%
  mutate(Age_bin = cut(Age, breaks = 5, include.lowest = TRUE)) %>%
  count(Age_bin) %>%
  ggplot(aes(x = Age_bin, y = n)) +
  geom_col() +
  labs(
```

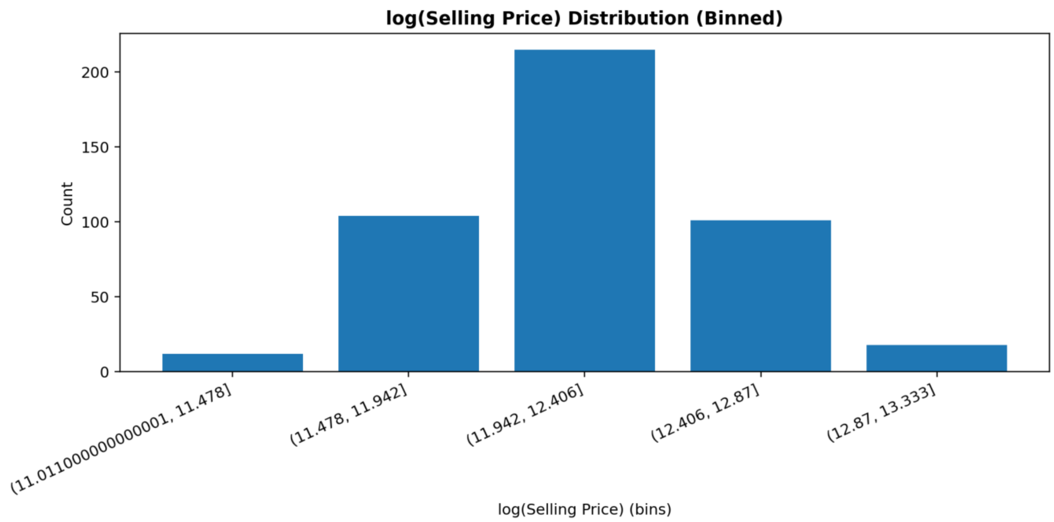
```
    title = "Age Distribution (Binned)",
    x = "Age (bins)",
    y = "Count"
  ) +
  portfolio_theme() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1))

# OtherArea (binned)
datos %>%
  mutate(OtherArea_bin = cut(OtherArea, breaks = 5, include.lowest = TRUE)) %>%
  count(OtherArea_bin) %>%
  ggplot(aes(x = OtherArea_bin, y = n)) +
  geom_col() +
  labs(
    title = "Other Area Distribution (Binned)",
    x = "Other Area (bins)",
    y = "Count"
  ) +
  portfolio_theme() +
  theme(axis.text.x = element_text(angle = 25, hjust = 1))
```



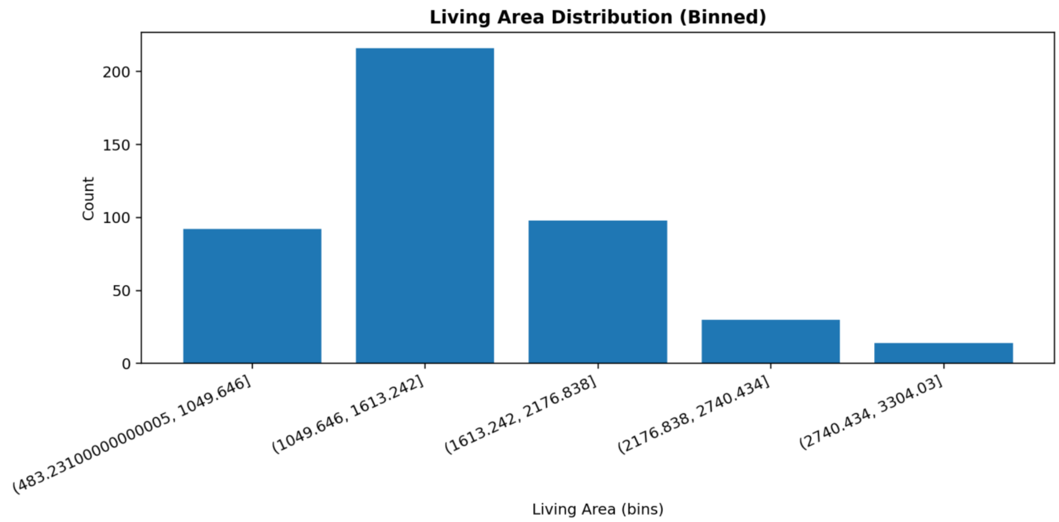
Interpretation: Latitude bins indicate how observations are distributed north-south. Strong imbalance can bias spatial models toward dense areas.



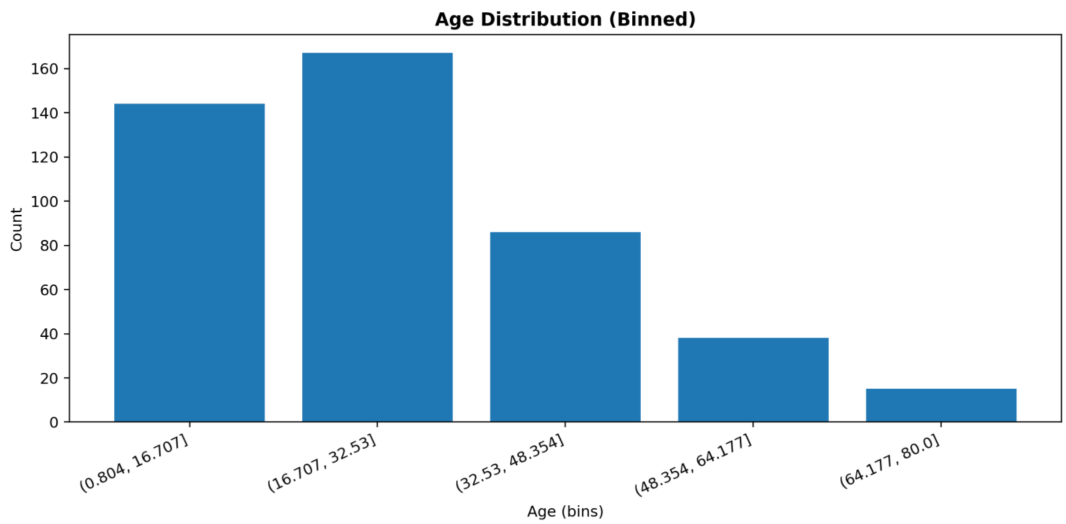
Interpretation: Longitude bins indicate east-west coverage. Gaps may appear as extrapolation zones in kriging surfaces.



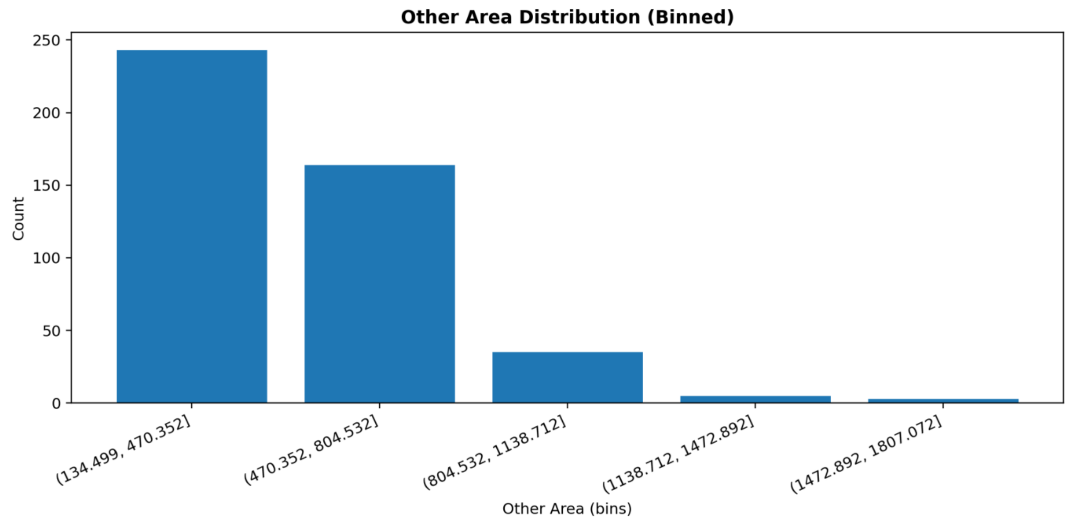
Interpretation: Binning log price gives a coarse view of market segmentation (low, mid, high).



Interpretation: Binned living area shows the concentration of mid-size vs. large properties.



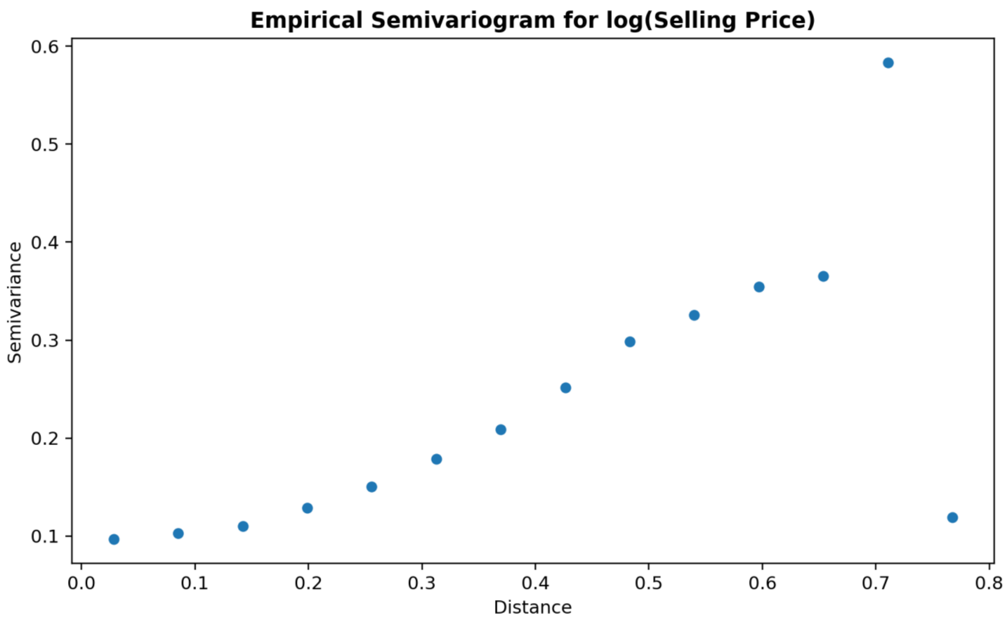
Interpretation: Age bins reveal whether the dataset is dominated by newer developments or older neighborhoods.



Interpretation: Other area bins help assess how common auxiliary spaces are across the sample.

1.5) Empirical Semivariogram

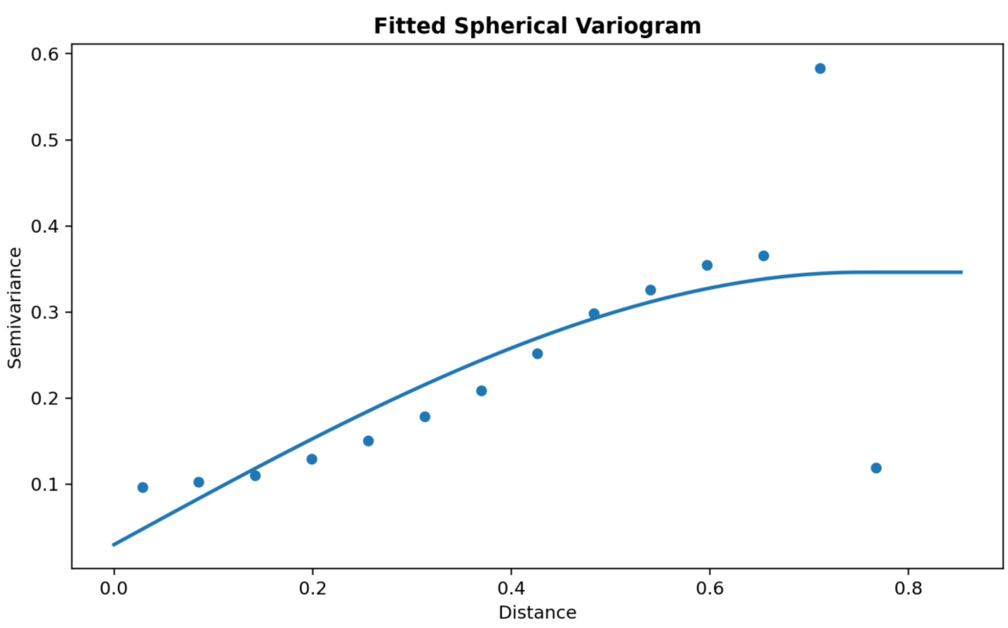
```
# -------------------------
# Empirical semivariogram
# -------------------------

# Convert to SpatialPointsDataFrame (required for gstat)
datos_sp <- datos
coordinates(datos_sp) <- c("Longitude", "Latitude")

variograma <- variogram(logSellingPr ~ 1, datos_sp)

# Empirical variogram
plot(
  variograma,
  main = "Empirical Semivariogram for log(Selling Price)",
  xlab = "Distance",
  ylab = "Semivariance",
  pch = 16
)
```



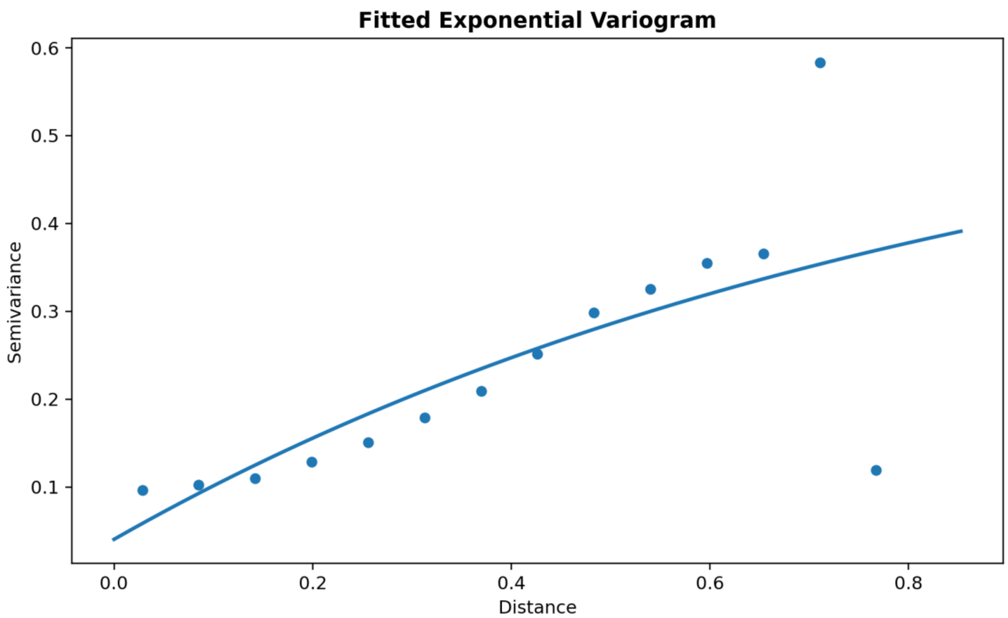
Interpretation: The empirical semivariogram summarizes spatial autocorrelation: semivariance increases with distance when nearby locations have more similar prices than far-apart locations. This guides selection of a variogram model for kriging.

1.6) Variogram Model Fitting

```
# -------------------------
# Fitted variogram models
# -------------------------

# Spherical
variograma_spherical <- vgm(psill = 0.09, model = "Sph", range = 0.15, nugget = 0.03)
ajuste_spherical <- fit.variogram(variograma, variograma_spherical, fit.sills = TRUE, fit.ranges = FALSE)

plot(
  variograma,
  model = ajuste_spherical,
  main = "Fitted Spherical Variogram",
  xlab = "Distance",
  ylab = "Semivariance",
  pch = 16,
  lwd = 2,
  lty = 1
)

# Exponential
variograma_exponential <- vgm(psill = 0.08, model = "Exp", range = 0.15, nugget = 0.06)
ajuste_exponential <- fit.variogram(variograma, variograma_exponential, fit.sills = TRUE, fit.ranges = TRUE)

plot(
  variograma,
  model = ajuste_exponential,
  main = "Fitted Exponential Variogram",
  xlab = "Distance",
  ylab = "Semivariance",
  pch = 16,
  lwd = 2,
  lty = 1
)
```



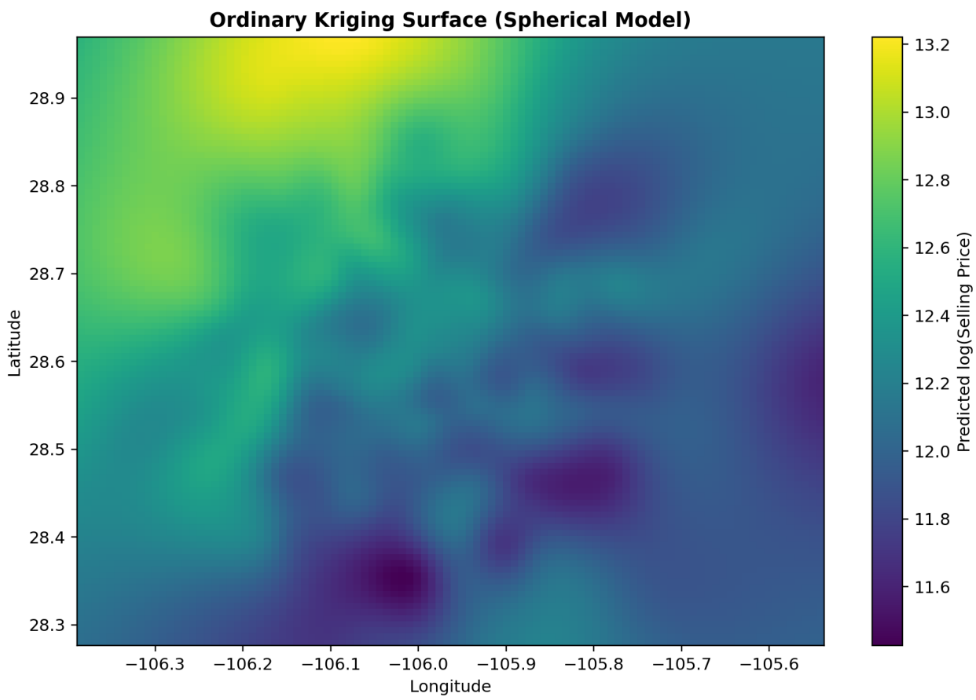
Interpretation: The spherical model implies correlation weakens until reaching a finite range, after which locations are effectively uncorrelated. It is common in regionalized variables.



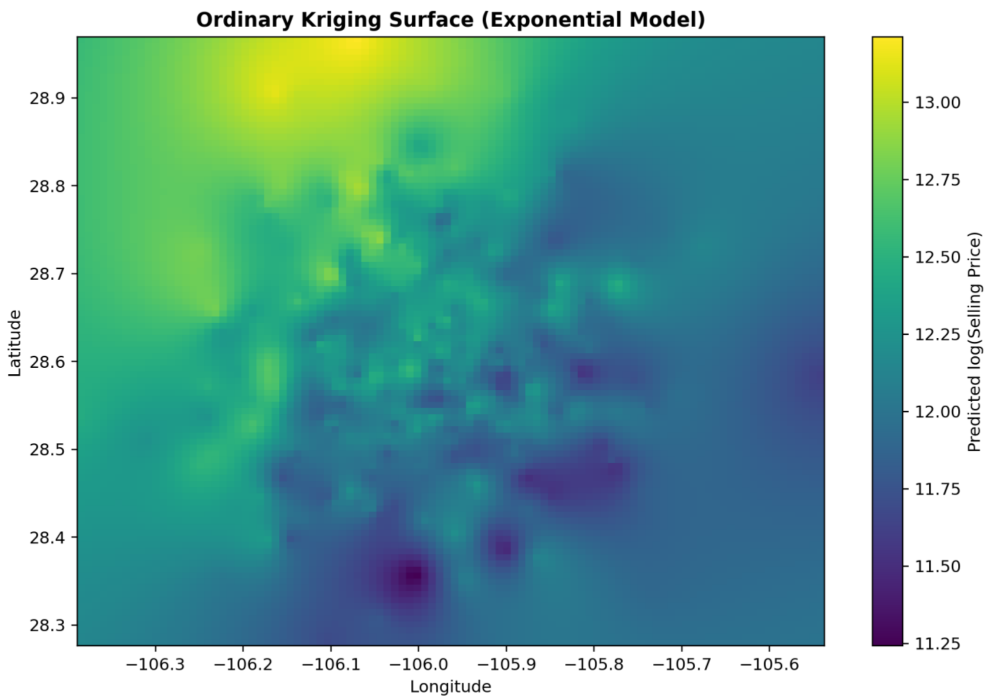
Interpretation: The exponential model decays more gradually and does not reach a strict sill at a finite distance, often capturing smoother long-range spatial dependence.

1.7) Kriging Surfaces

```
# -------------------------
# Kriging maps (spherical vs exponential)
# -------------------------

# Grid
bbox_vals <- bbox(datos_sp)

grilla <- expand.grid(
  Longitude = seq(bbox_vals["Longitude", "min"], bbox_vals["Longitude", "max"], length.out = 100),
  Latitude  = seq(bbox_vals["Latitude",  "min"], bbox_vals["Latitude",  "max"], length.out = 100)
)
coordinates(grilla) <- ~ Longitude + Latitude

# Use fitted models when available; fall back to nominal if fit fails
sph_model <- if (inherits(ajuste_spherical, "variogramModel")) ajuste_spherical else vgm(psill = 1, model = "Sp
exp_model <- if (inherits(ajuste_exponential, "variogramModel")) ajuste_exponential else vgm(psill = 1, model =

kriging_sph <- gstat(formula = logSellingPr ~ 1, locations = datos_sp, model = sph_model)
kriging_exp <- gstat(formula = logSellingPr ~ 1, locations = datos_sp, model = exp_model)

sph_result <- predict(kriging_sph, newdata = grilla)
exp_result <- predict(kriging_exp, newdata = grilla)

# Tidy for ggplot
grilla_df <- as.data.frame(grilla)
grilla_df$sph_prediction <- sph_result$var1.pred
grilla_df$exp_prediction <- exp_result$var1.pred

# Map: Spherical
ggplot(grilla_df, aes(x = Longitude, y = Latitude, fill = sph_prediction)) +
  geom_tile() +
  coord_equal() +
  scale_fill_viridis_c(labels = label_number(accuracy = 0.01)) +
  labs(
    title = "Ordinary Kriging Surface (Spherical Model)",
    x = "Longitude",
    y = "Latitude",
    fill = "Predicted log(Selling Price)"
  ) +
  portfolio_theme()

# Map: Exponential
ggplot(grilla_df, aes(x = Longitude, y = Latitude, fill = exp_prediction)) +
  geom_tile() +
  coord_equal() +
  scale_fill_viridis_c(labels = label_number(accuracy = 0.01)) +
  labs(
    title = "Ordinary Kriging Surface (Exponential Model)",
    x = "Longitude",
    y = "Latitude",
    fill = "Predicted log(Selling Price)"
  ) +
  portfolio_theme()
```



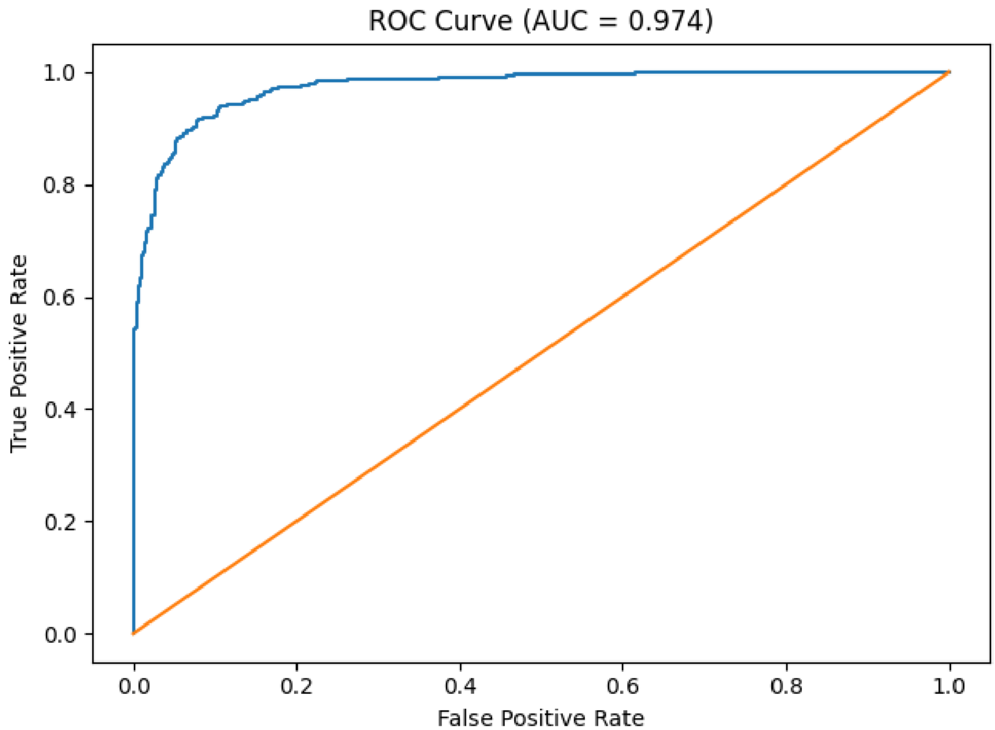
Interpretation: The spherical kriging surface highlights spatial gradients and local pockets of higher or lower predicted log price. It is suitable when dependence fades after a finite range.



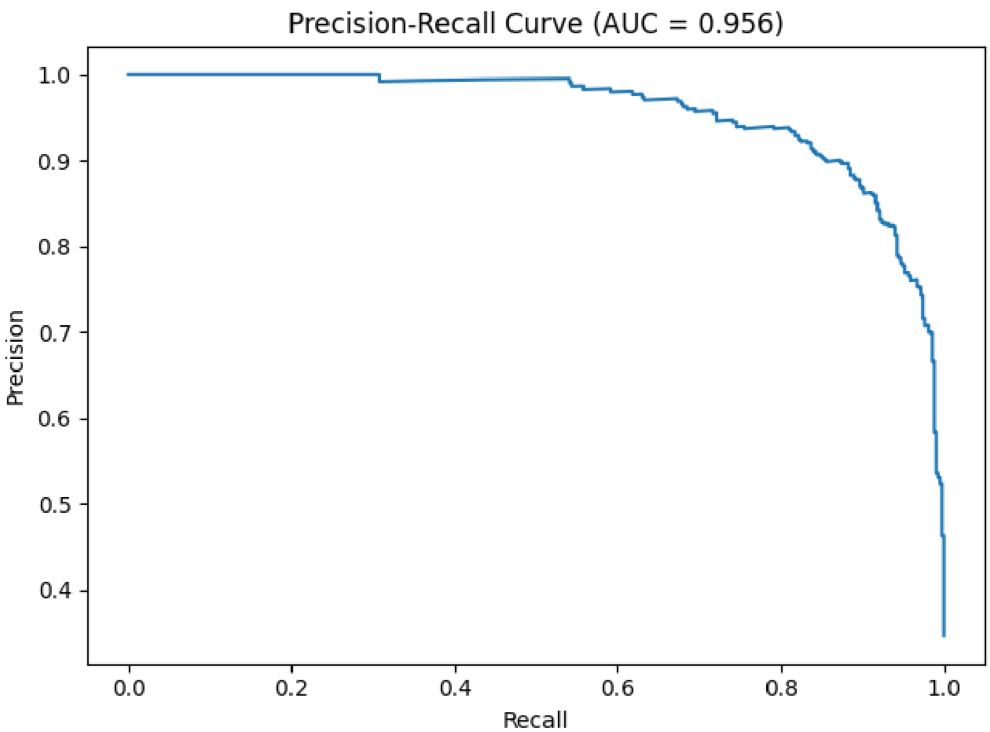
Interpretation: The exponential kriging surface tends to appear smoother, reflecting longer-range correlation. Comparing both maps helps assess sensitivity to the chosen variogram model.

2) Model Diagnostic Panels (Provided Figures)

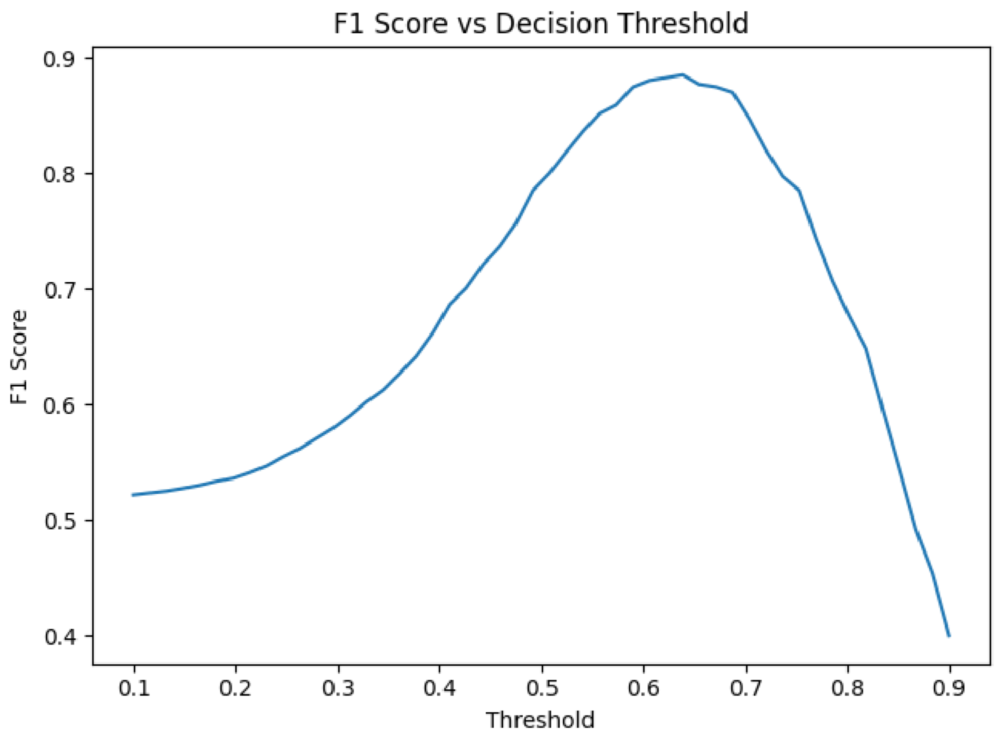
The following figures are included as additional model evaluation evidence (ROC, PR, threshold tuning, feature importance, and confusion matrices).



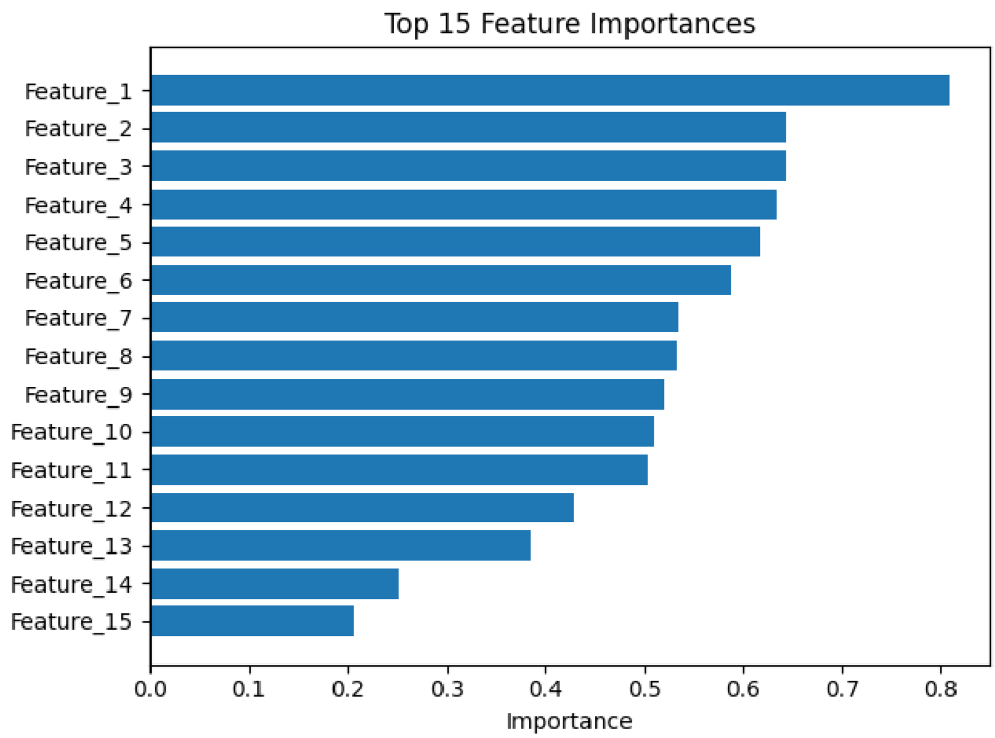
Interpretation: ROC Curve: Plots true positive rate vs. false positive rate across thresholds. AUC closer to 1 indicates strong ranking performance.



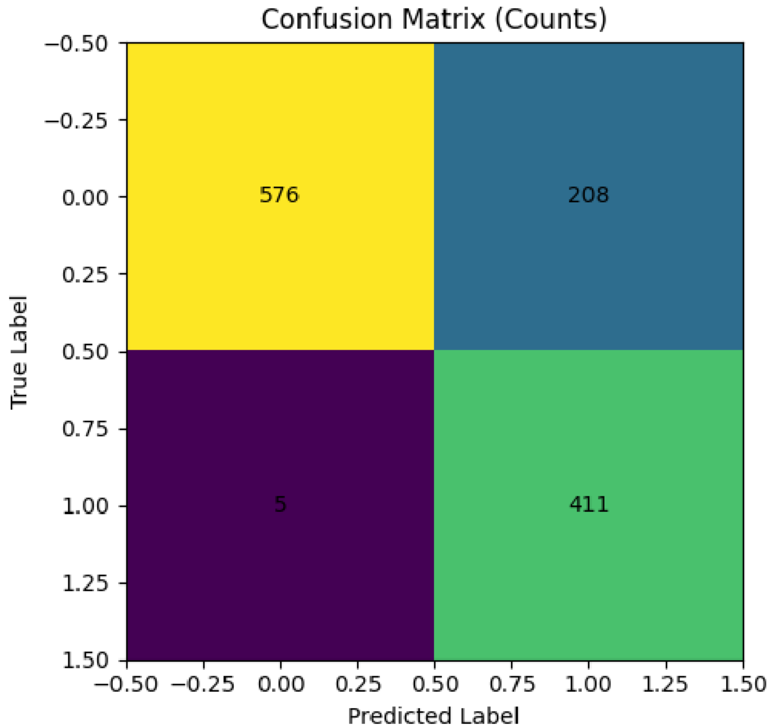
Interpretation: Precision-Recall Curve: Highlights performance under class imbalance. AUC summarizes the trade-off between precision and recall.



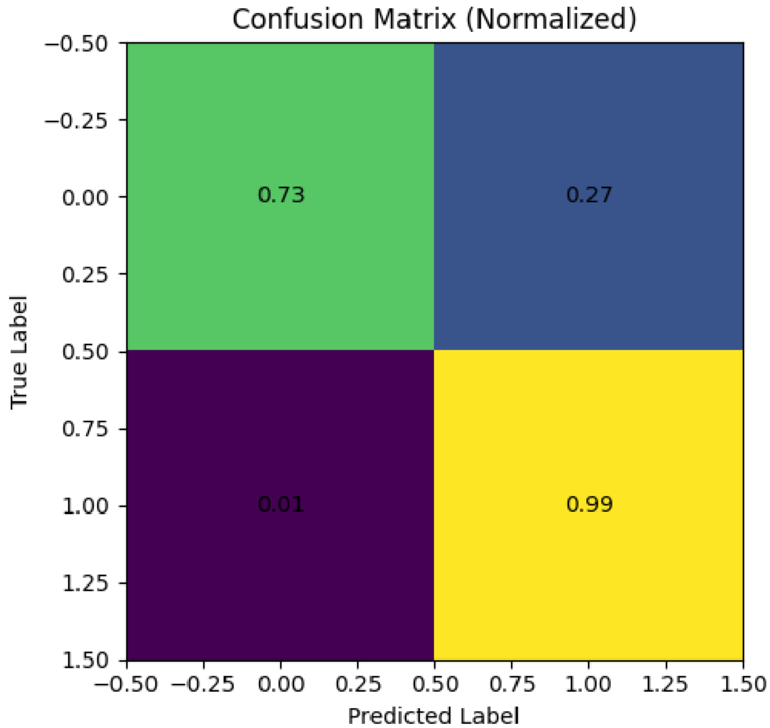
Interpretation: F1 vs Threshold: Shows the decision threshold that maximizes the balance between precision and recall.



Interpretation: Top Feature Importances: Ranks predictors by contribution to the model, supporting interpretability and stakeholder communication.



Interpretation: Confusion Matrix (Counts): Raw classification outcomes (TN, FP, FN, TP). Useful for operational impact assessment.



Interpretation: Confusion Matrix (Normalized): Row-normalized rates by true class, making error patterns comparable across classes.