Clasificación de Niveles Socioeconómicos en el Área Metropolitana de Lima usando Machine Learning

Análisis basado en Imágenes de Google Street View

Michael Hinojosa Jorge Quenta Mikel Bracamonte Eduardo Aragon

Facultad de Computación
Universidad de Ingeniería y Tecnología

Noviembre 2025

Contenido

1. Introducción
2. Objetivos
3. Related Work
4. Metodología
5. Feature Extraction
6. Modelos
7. EDA
8. Hiperparámetros
9. Resultados
10. Conclusiones

Contexto del Problema

Problemática

- Los censos nacionales se realizan cada 10 años
- Lima Metropolitana tiene $>$ 10 millones de habitantes
- Crecimiento demográfico rápido y transformaciones económicas constantes
- Necesidad de métodos más frecuentes de monitoreo socioeconómico

Solución Propuesta

Utilizar Machine Learning e imágenes de Google Street View para predecir niveles socioeconómicos (Alto, Medio, Bajo) en vecindarios de Lima Metropolitana.

Objetivos del Proyecto

Objetivo Principal

Estudiar la clasificación de niveles socioeconómicos en el Área Metropolitana de Lima mediante el entrenamiento de modelos de machine learning con 15,000 imágenes de diferentes distritos.

Objetivos Específicos

1. Crear un dataset de 15,000 imágenes del área metropolitana y etiquetarlas por categoría
2. Extraer vectores de características usando DINOv2
3. Entrenar cuatro modelos de machine learning con el dataset
4. Evaluar los modelos usando métricas: Accuracy, Precision, Recall y F1-Score

Trabajos Relacionados

Clasificación Urbana con Imágenes Satelitales

Rahman et al. (2021)

- Usaron imágenes de Google Earth (50×50 km)
- Modelo DeepLabv3+ en 7 ciudades
- Lima: 96.75 % accuracy (el más alto)
- Categorizaron en 4 clases socioeconómicas
- Demostraron viabilidad de clasificación con imágenes

Predicción desde Street-View

Machicao et al. (2022)

- Usaron Google Street View en Brasil
- Región semi-rural (Vale do Ribeira)
- Feature extractor: VGG-16
- 5 clases de ingreso: 55 % accuracy exacta
- 80 % accuracy con tolerancia de 1 clase
- Mejor predicción en clase alta (80 %)

Nuestro Aporte

Combinamos el enfoque urbano de Rahman con las imágenes street-level de Machicao, aplicando DINOv2 (embeddings más potentes que VGG-16) en un entorno altamente urbanizado: Lima Metropolitana.

Metodología

Fuente de Datos:
Estudio INEI 2020 - Niveles socioeconómicos a nivel de manzana

Simplificación de Categorías:

- Alto: Alto
- Medio: Medio Alto + Medio
- Bajo: Medio Bajo + Bajo

Distritos Seleccionados (12):
Alto: Miraflores, San Isidro, La Molina, San Borja
Medio: La Victoria, Breña, Lince, Los Olivos
Bajo: Carabayllo, SJL, Villa El Salvador, VMT

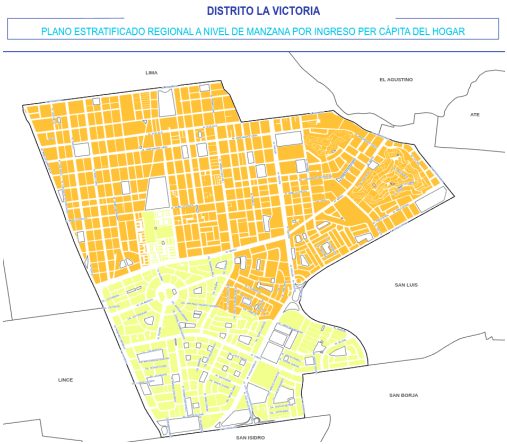

Figura: Estratificación en La Victoria

Total: 15,000 imágenes de Google Street View

Extracción de Características: DINOv2

¿Qué es DINOv2?

Self-Distillation with No Labels v2 - Modelo de Meta AI basado en Vision Transformers (ViT)

Características del Modelo:

- Pre-entrenado con imágenes sin etiquetas
- Aprendizaje auto-supervisado
- Captura features de bajo nivel (texturas, colores) y alto nivel (objetos, escenas)

Arquitectura Utilizada:

- DINOv2-giant (ViT-g/14)
- Input: 224×224 píxeles
- Output: 1536 dimensiones
- Aceleración con CUDA

Los vectores de características sirven como input para los modelos de clasificación

Modelos Base y Adicionales

- Logistic Regression: Modelo base lineal con estrategia One-vs-Rest (OvR)
- Support Vector Machine (SVM): Kernel RBF para capturar patrones no lineales
- Multilayer Perceptron (MLP): Red neuronal con 3 capas ocultas y activación ReLU

XGBoost (Modelo Moderno)

Extreme Gradient Boosting - Método ensemble basado en árboles de decisión
Predicción como suma de K árboles:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

Función objetivo con regularización:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

donde $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$

Análisis Exploratorio de Datos - Distribución y Calidad

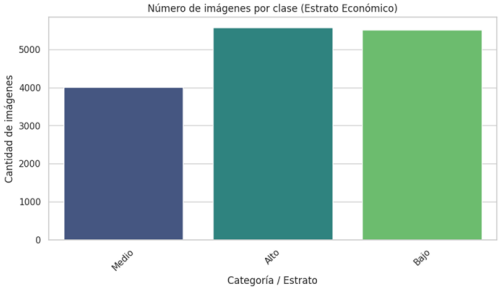

Figura: Distribución balanceada

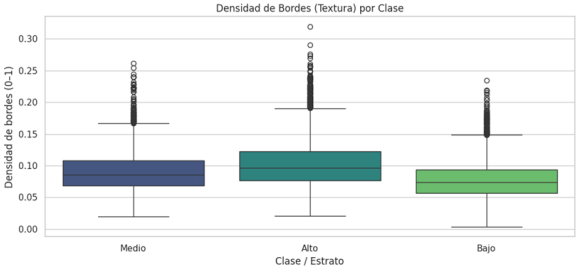

Figura: Densidad de bordes por clase

Calidad del Dataset:

- Sin archivos corruptos
- Resolución uniforme (640×640)
- Dataset limpio y listo

Análisis de Textura:

- Alto: Mayor densidad (más detalles arquitectónicos)
- Medio: Densidad intermedia
- Bajo: Menor densidad (superficies planas)

Análisis Exploratorio - Patrones Visuales

Figura: Mosaicos: Alto, Medio, Bajo

Observaciones Cualitativas:

- Alto: Calles mantenidas, vegetación abundante, arquitectura moderna
- Medio: Calles pavimentadas, edificios uniformes, vegetación moderada
- Bajo: Calles sin pavimentar, construcciones informales, poca vegetación

Análisis de Color (RGB):

Clase	R	G	B
Alto	135.4	135.6	128.4
Medio	136.7	136.2	130.9
Bajo	147.5	144.0	135.5

- Clase Baja: RGB más altos
- Clases Alta/Media similares
- *Color solo* insuficiente

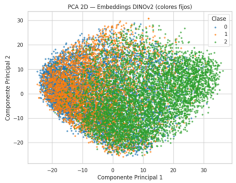

Figura: PCA 2D: No separa clases

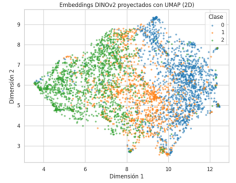

Figura: UMAP 2D: Mejor separación

Métodos Lineales:

- PCA: No separa clases
- $<15\,\%$ varianza en 2 componentes
- 50 componentes para $>60\,\%$ varianza

Métodos No Lineales:

- t-SNE: Clusters parciales con overlap
- UMAP: Mejor separación visual
- ISOMAP: Estructura geodésica continua

Conclusión: Las clases siguen un *gradiente continuo* en espacio no lineal, requiriendo modelos con fronteras complejas.

Análisis Exploratorio - Visualización 3D

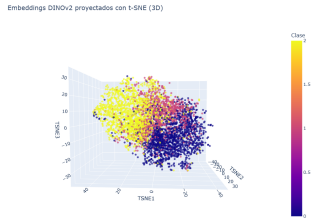

Figura: t-SNE 3D

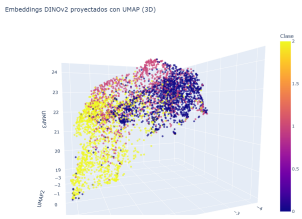

Figura: UMAP 3D

t-SNE 3D:

- Mejor separación que en 2D
- Clase 0 y 2 en regiones distintas
- Clase 1 posicionada entre ambas
- Overlap moderado persiste

UMAP 3D:

- La mejor separación observada
- Tres grupos parcialmente diferenciados
- Clase 1 actúa como puente
- Preserva estructura local y global

Conclusión: Las visualizaciones 3D confirman que existe estructura discriminativa en los embeddings, con UMAP revelando la organización más clara del espacio latente.

Análisis de Hiperparámetros - Modelos Base

Logistic Regression

- Pipeline: StandardScaler + OvR
- Parámetros: $C \in \{0{,}01, 0{,}1, 1, 10\}$
- Validación: 5-fold CV
- Mejor: $C = 0{,}01$ (L2)
- Test: 89.66 %

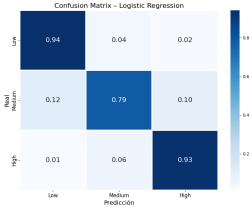

Figura: Confusion Matrix

Support Vector Machine

- Búsqueda: RandomizedSearchCV (50 iter)
- Kernel: RBF
- Parámetros: C, γ
- Mejor: $C = 2{,}69$, $\gamma = 0{,}00048$
- Test: 91.13 %

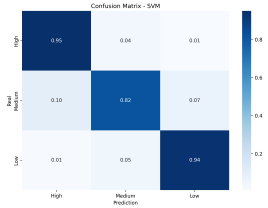

Figura: Confusion Matrix - Mejor resultado

Análisis de Hiperparámetros - Modelos Avanzados

Multilayer Perceptron (MLP)

- Arquitectura: 1536 $\rightarrow$ hidden $\rightarrow$ 3
- Validación: 3-fold CV
- Parámetros explorados:
  - Hidden: [1024,512,256], [768,384,192]
  - LR: 10^{-3}, 510^{-4}
  - Batch: 256, 512
  - Dropout: 0.2
- Mejor: [768,384,192], lr=0.001, batch=256
- Test: 89.93 %

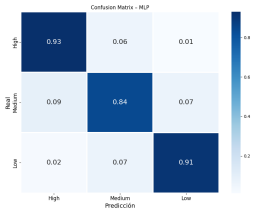

Figura: Confusion Matrix

XGBoost

- Pipeline: StandardScaler + XGBClassifier
- Parámetros: max_depth $\in \{6, 8\}$
- learning_rate: 0.1
- Estimadores: 250 árboles
- Mejor: max_depth=6
- Test: 88.31 %

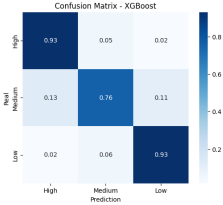

Figura: Confusion Matrix

Resultados Comparativos

Cuadro: Métricas de clasificación para los cuatro modelos

Modelo	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.8966	0.8900	0.8867	0.8867
SVM	0.9113	0.9106	0.9113	0.9107
MLP	0.8993	0.9000	0.8993	0.8995
XGBoost	0.8800	0.8767	0.8767	0.8800

Observaciones Principales

- SVM obtiene el mejor F1-Score general (91.07 %)
- MLP segundo lugar (89.95 %), seguido por Logistic Regression (88.67 %)
- XGBoost obtiene el menor F1-Score (88.00 %), probablemente por alta dimensionalidad
- Todos los modelos muestran F1-Scores altos en clases Alto y Bajo (>0.90)
- La clase Medio es la más desafiante (F1 $\sim$0.79-0.85) por overlap natural
- Los embeddings DINOv2 demuestran ser altamente efectivos para clasificación balanceada

Conclusiones

Hallazgos Principales

1. SVM logró el mejor desempeño: 91.13 % accuracy usando kernel RBF, seguido por MLP (89.93 %), Logistic Regression (89.66 %), y XGBoost (88.31 %)
2. Estructura no lineal: Las clases siguen un manifold continuo, no clusters separados — esto explica el éxito de modelos no lineales
3. DINOv2 fue crucial: Los embeddings de 1536 dimensiones capturaron patrones de bajo y alto nivel efectivamente
4. Clase Media más difícil: F1-scores de 0.79-0.85 vs. 0.89-0.95 para Alto/Bajo, debido a su naturaleza transicional
5. XGBoost mostró menor desempeño, posiblemente por la alta dimensionalidad (1536 features) que no favorece a métodos basados en árboles

Limitaciones y Trabajo Futuro

Limitaciones: Solo 12 de 43 distritos cubiertos; datos INEI 2020 pueden estar desactualizados; simplificación a 3 categorías perdió granularidad

Trabajo Futuro:

- Expandir dataset a todos los distritos de Lima
- Fine-tuning de DINOv2 para patrones específicos de Lima
- Aprendizaje multi-modal (street-view + satélite + OSM + censo)
- Análisis temporal para monitorear cambios urbanos
- Transfer learning a otras ciudades latinoamericanas

Gracias por su atención

¿Preguntas?

Código disponible en:
`https://github.com/m41k1204/ml_NSE_classifier`