

METODOLOGÍA - RESUMEN EJECUTIVO



Fuente de Datos y Preparación

Dataset Utilizado:

- Fuente: Archivo Nacional de Datos de Colombia (ANDA) - Estadísticas Vitales EEVV
- Período: 2009 a 2019
- Registros: 642,657 nacimientos
- Variables: 23 columnas con información demográfica y de salud

Preprocesamiento de Datos:

Se siguió la metodología CRISP-DM con los siguientes pasos:

1. Limpieza inicial:
 - Eliminación de columnas irrelevantes (AREANAC, SIT_PARTO, AREA_RES, SEG_SOCIAL, IDPERTET)
 - Remoción de datos duplicados
 - Completado de valores faltantes con el promedio entre valores adyacentes
 - Eliminación de registros que permanecían con valores nulos
2. Transformación:
 - Normalización de datos con método Min-Max (escala 0-1)
 - Análisis exploratorio de distribuciones y patrones
 - Estudio de correlaciones entre variables



Método de Ensamble Propuesto

Algoritmos Base:

- Random Forest (principal)
- Árboles de Decisión
- Regresión Lineal
- k-NN (k-Nearest Neighbors)

Técnica de Ensamble: BAGGING

(Bootstrap Aggregating)

Implementación en 4 Pasos:

Paso 1 - Muestreo Bootstrap:

text

Se crean L subconjuntos del dataset original:

$\{Z_1^1, Z_2^1, Z_{R^1}\}, \{Z_1^2, Z_2^2, Z_{R^2}\}, \dots, \{Z_1^L, Z_2^L, Z_{R^L}\}$

- Muestras con reemplazo
- Tamaño igual al conjunto original

Paso 2 - Modelado Individual:

text

Se entrena un modelo en cada subconjunto:

$W_1, W_2, \dots, W_L \rightarrow f_1(x), f_2(x), \dots, f_L(x)$

Paso 3 - Aprendizaje Paralelo:

- Cada modelo aprende independientemente
- Entrenamiento concurrente en todos los subconjuntos

Paso 4 - Combinación de Predicciones:

text

Predicción final = Promedio de todas las predicciones:

$f(x) = \sum f_i(x) / L \quad \text{para } i = 1 \text{ hasta } L$

Variables Clave Analizadas

Variables Predictoras Seleccionadas:

1. EDAD_MADRE - Edad de la madre al parto
2. NIV_EDUM - Nivel educativo de la madre

3. NIV_EDUP - Nivel educativo del padre
4. EST_CIVM - Estado civil de la madre
5. EDAD_PADRE - Edad del padre
6. N_HIJOSV - Número de hijos nacidos vivos

Variable Objetivo:

- Número de hijos por madre (comportamiento reproductivo)



Evaluación del Modelo

Métricas de Performance:

- R^2 (Coeficiente de Determinación):
 - Mide qué tan bien se ajustan los datos a la línea de regresión
 - Rango: 0 a 1 (mejor entre más cercano a 1)
- RMSE (Raíz del Error Cuadrático Medio):
 - Representa diferencia entre valores reales y predichos
 - Mejor entre más cercano a 0

Estrategia de Validación:

- División 70%-30%: 70% entrenamiento, 30% testing
- Validación cruzada para estimar performance
- Comparación con modelos individuales



Análisis de Importancia de Variables

Técnica Utilizada:

- Importancia por Permutación
- Validación cruzada para estimar influencia
- Matriz de correlación para relaciones lineales

Variables Más Influyentes:

1. Edad de la madre (42.3% de importancia)
2. Nivel educativo madre (34.9%)
3. Nivel educativo padre (10.4%)
4. Estado civil madre (4.1%)
5. Edad del padre (3.4%)



Ventajas del Enfoque Propuesto

Reducción de Varianza:

- Bagging combina múltiples estimaciones
- Compensación de errores entre modelos
- Mayor estabilidad en predicciones

Mejora de Precisión:

- 20% mejor RMSE vs árboles de decisión simples
- 30% mejor RMSE vs regresión lineal clásica
- 37-48% mejora en R^2 vs métodos individuales

Esta metodología demostró ser efectiva para identificar factores clave en el comportamiento reproductivo y puede servir como referencia para estudios similares en otros contextos demográficos.

METODOLOGÍA

3.1. Fuente de Datos y Preparación

3.1.1. Dataset Utilizado

Para el desarrollo de esta investigación, se empleará el dataset "Registros de Nacidos Vivos en el Perú (2015-2025)" administrado por el Ministerio de Salud (MINSA). Este conjunto de datos contiene información detallada proveniente del Certificado de Nacido Vivo (CNV) con las siguientes características:

- Período: 2015 a 2025 (datos trimestrales)
- Variables disponibles: 22 atributos demográficos, de salud y socioculturales
- Cobertura: Nacional y desagregada por departamentos
- Actualización: Trimestral

3.1.2. Preprocesamiento de Datos

Se seguirá la metodología CRISP-DM con los siguientes procesos:

Limpieza inicial:

- Eliminación de registros duplicados y inconsistentes
- Validación de rangos de valores (ej: edad materna entre 12-55 años)
- Manejo de valores faltantes mediante interpolación temporal
- Estandarización de formatos de fecha y categorías

Selección de variables:

Se conservarán las variables más relevantes basadas en el estudio colombiano y la disponibilidad en el dataset peruano:

- FecNac_Año, FecNac_Mes (para series temporales)
- Edad_Madre, Nivel_Instrucción_Madre
- ubiLbIgeoInel (para análisis departamental)
- Estado_Civil, Num_embor_madre, Hijos_vivo_madre

Transformación de datos:

- Agregación trimestral de nacimientos
- Codificación de variables categóricas (nivel educativo, estado civil)

- Normalización Min-Max para variables numéricas
- Creación de variables dummy para estacionalidad

3.2. Enfoque de Modelado

3.2.1. Algoritmos Base

Inspirado en el éxito del estudio colombiano, se implementarán los siguientes algoritmos:

Modelos Individuales:

- Regresión Lineal Múltiple: Como línea base para comparación
- Random Forest: Para capturar relaciones no lineales
- k-NN Regressor: Para patrones basados en similitud
- Decision Trees: Como componente base del ensamble

3.2.2. Método de Ensamble Bagging

Se implementará la técnica de Bootstrap Aggregating siguiendo el mismo enfoque exitoso del estudio colombiano:

Paso 1: Muestreo Bootstrap

text

Creación de L subconjuntos de entrenamiento:

$\{D_1, D_2, \dots, D_L\}$ donde cada D_i se genera mediante muestreo con reemplazo del dataset original

Paso 2: Entrenamiento Paralelo

- Cada subconjunto D_i entrena un modelo de Random Forest
- Configuración: 100 estimadores, profundidad máxima 10
- Parámetros optimizados mediante validación cruzada

Paso 3: Generación de Predicciones

Cada modelo produce predicciones para:

- Tasa de natalidad trimestral nacional
- Tasa de natalidad trimestral por departamento

Paso 4: Combinación por Promedio

text

Predicción final = $(1/L) * \sum \text{predicción_modelo_i}$

3.2.3. Adaptaciones Específicas para el Contexto Peruano

Variables de Entrada Especializadas:

text

X = [Edad_Madre, Nivel_Educativo_Madre, Trimestre, Año,
Codigo_Departamento, Tendencia_Temporal, Variables_Estandarizadas]

Salidas del Modelo:

- Y_nacional = Tasa de natalidad trimestral nacional
- Y_departamental = Tasa de natalidad trimestral por departamento

3.3. Configuración Experimental

3.3.1. División de Datos

- Entrenamiento: 2015-2019 (70% del período)
- Validación: 2020-2022 (20% del período)
- Testing: 2023-2025 (10% del período)

3.3.2. Métricas de Evaluación

Las mismas utilizadas en el estudio colombiano para permitir comparación:

Coeficiente de Determinación (R^2):

text

$R^2 = 1 - (\sum(y_{\text{real}} - y_{\text{pred}})^2 / \sum(y_{\text{real}} - y_{\text{promedio}})^2)$

- Interpretación: Proporción de varianza explicada
- Rango: 0 a 1 (mejor cerca de 1)

Raíz del Error Cuadrático Medio (RMSE):

text

$$\text{RMSE} = \sqrt{(1/n * \sum (y_{\text{real}} - y_{\text{pred}})^2)}$$

- Interpretación: Error promedio en unidades originales
- Rango: ≥ 0 (mejor cerca de 0)

3.3.3. Validación

- Validación Cruzada Temporal: TimeSeriesSplit con 5 folds
- Validación de Estabilidad: Múltiples ejecuciones con diferentes semillas
- Análisis de Residuales: Para detectar patrones no capturados

3.4. Análisis de Importancia de Variables

3.4.1. Técnica de Importancia por Permutación

- Permutación aleatoria de cada variable predictora
- Medición del decremento en performance
- Validación cruzada para robustez estadística

3.4.2. Análisis Esperado de Variables

Basado en el estudio colombiano, se anticipa:

1. Edad de la madre (factor principal)
2. Nivel educativo materno
3. Ubicación geográfica (diferencias departamentales)
4. Factores estacionales (trimestres)

3.5. Ventajas del Enfoque Propuesto

3.5.1. Robustez

- Reducción de varianza mediante combinación de múltiples modelos

- Compensación de errores entre predictores individuales
- Estabilidad en predicciones a largo plazo

3.5.2. Adaptabilidad

- Capacidad de capturar patrones no lineales
- Adaptación a cambios estructurales en tendencias
- Escalabilidad para análisis multinivel (nacional/departamental)

3.5.3. Interpretabilidad

- Importancia de variables cuantificable
- Análisis de contribuciones marginales
- Transparencia en el proceso predictivo

3.6. Implementación Técnica

3.6.1. Herramientas

- Lenguaje: Python 3.9+
- Librerías: Scikit-learn, Pandas, NumPy, Matplotlib
- Procesamiento: Google Colab / Jupyter Notebooks

3.6.2. Flujo de Procesamiento

text

Datos Crudos → Limpieza → Agregación Trimestral →
Entrenamiento Modelos → Ensamble Bagging →

Validación → Análisis Resultados

Esta metodología, basada en el enfoque probado exitosamente en Colombia pero adaptada al contexto y datos peruanos, permitirá desarrollar un modelo robusto y preciso para la predicción de tasas de natalidad, proporcionando así una herramienta valiosa para la planificación de políticas públicas en el Perú.



EXPLICACIÓN DETALLADA: VALIDACIÓN DE RANGOS DE VALORES

¿Qué significa "Validación de rangos de valores"?

Es un proceso de control de calidad donde verificamos que los valores de cada variable estén dentro de límites lógicos y biológicamente posibles. Es como un "filtro de sentido común" para los datos.



EJEMPLO CONCRETO: EDAD MATERNA

Límites propuestos: 12-55 años


- Límite inferior (12 años): Edad biológica mínima para un embarazo viable
- Límite superior (55 años): Edad máxima común para embarazos naturales

¿Qué hacemos si encontramos valores fuera de rango?


Caso 1: Edad = 8 años

-  BIOLÓGICAMENTE IMPOSIBLE
- Acción: Eliminar registro (error de digitación)

Caso 2: Edad = 65 años

-  POCO PROBABLE
- Acción: Verificar si es error o caso excepcional real

Caso 3: Edad = -1 o 999

-  VALORES POR DEFECTO/ERROR
 - Acción: Eliminar como dato corrupto
-



OTROS EJEMPLOS EN TU DATASET

Peso del recién nacido (PESO_NACIDO)

- Rango válido: 500 - 5,000 gramos
- Fuera de rango: 50g (inviabile) o 10,000g (error)

Talla del recién nacido (TALLA_NACIDO)

- Rango válido: 30 - 60 cm
- Fuera de rango: 10 cm o 200 cm (imposible)

Duración del embarazo (DJ/L_EMB_PARTO)

- Rango válido: 20 - 45 semanas
- Fuera de rango: 10 semanas (aborto) o 60 semanas (error)

CÓMO LO IMPLEMENTARÍAS EN CÓDIGO

python

```
def validar_rangos(datos):  
    # Edad materna  
    datos = datos[(datos['Edad_Madre'] >= 12) & (datos['Edad_Madre'] <= 55)]  
  
    # Peso del bebé  
    datos = datos[(datos['PESO_NACIDO'] >= 500) & (datos['PESO_NACIDO'] <= 5000)]  
  
    # Talla del bebé  
    datos = datos[(datos['TALLA_NACIDO'] >= 30) & (datos['TALLA_NACIDO'] <= 60)]  
  
    # Duración embarazo  
    datos = datos[(datos['DJ/L_EMB_PARTO'] >= 20) & (datos['DJ/L_EMB_PARTO'] <= 45)]  
  
    return datos  
  
# Aplicar validación
```

```
datos_limpios = validar_rangos(datos_originales)
```

¿POR QUÉ ES IMPORTANTE ESTE PASO?

1. Calidad del Modelo

- Datos incorrectos → Predicciones incorrectas
- Los valores extremos distorsionan los patrones reales

2. Validez Científica

- Un modelo con datos biológicamente imposibles pierde credibilidad
- Garantiza que las conclusiones sean basadas en realidad

3. Ética de la Investigación

- Evita sacar conclusiones de casos que no existen en la realidad
 - Mantiene la integridad académica del trabajo
-



EN LA PRÁCTICA PARA TU TRABAJO

Ejemplo real que podrías encontrar:

- Registro con Edad_Madre = 9 años → Eliminar (error)
- Registro con PESO_NACIDO = 100g → Eliminar (inviable)
- Registro con Edad_Madre = 48 años → Mantener (caso real posible)

Resultado esperado:

- Dataset más pequeño pero más confiable
- Modelo que aprende de patrones reales
- Conclusiones válidas y aplicables

EXPLICACIÓN DETALLADA: TRANSFORMACIÓN DE DATOS

1. Agregación Trimestral de Nacimientos

¿Qué es?

Convertir los datos individuales de cada nacimiento en totales por trimestre.

¿Por qué se hace?

- Tu objetivo es predecir tasa trimestral, no nacimientos individuales
- Reduce la cantidad de datos manejable
- Permite análisis de tendencias temporales

Ejemplo práctico:

text

Datos originales:

- 15/01/2020: 1 nacimiento
- 20/01/2020: 1 nacimiento
- 10/02/2020: 1 nacimiento
- 05/03/2020: 1 nacimiento

→ Trimestre Q1-2020: 4 nacimientos

Implementación:

python

```
# Agrupar por año y trimestre
```

```
datos_trimestrales = datos.groupby(['Año', 'Trimestre']).agg({  
    'Nacimientos': 'count',  
    'Edad_Madre': 'mean',  
    'Nivel_Instruccion_Madre': 'mode'
```

```
}).reset_index()
```

2. Codificación de Variables Categóricas

¿Qué es?

Convertir texto en números que los algoritmos puedan entender.

Ejemplo: Nivel Educativo

```
text
```

```
Texto → Número
```

```
"Primaria incompleta" → 1
```

```
"Primaria completa" → 2
```

```
"Secundaria incompleta" → 3
```

```
"Secundaria completa" → 4
```

```
"Superior universitaria" → 5
```

Ejemplo: Estado Civil

```
text
```

```
"Soltera" → 0
```

```
"Conviviente" → 1
```

```
"Casada" → 2
```

```
"Divorciada" → 3
```

¿Por qué se hace?

- Los modelos matemáticos solo entienden números
 - Mantiene información ordinal (ej: más educación = número mayor)
-

3. Normalización Min-Max para Variables Numéricas

¿Qué es?

Escalar todas las variables numéricas al mismo rango (0 a 1).

Fórmula:

text

$$\text{valor_normalizado} = (\text{valor_original} - \text{mínimo}) / (\text{máximo} - \text{mínimo})$$

Ejemplo práctico: Edad de la madre

text

Edades originales: [15, 20, 25, 30, 35]

Mínimo = 15, Máximo = 35

15 → $(15-15)/(35-15) = 0.0$

20 → $(20-15)/(35-15) = 0.25$

25 → $(25-15)/(35-15) = 0.5$

30 → $(30-15)/(35-15) = 0.75$

35 → $(35-15)/(35-15) = 1.0$

¿Por qué se hace?

- Evita que variables con rangos grandes dominen el modelo
- Mejora la convergencia de los algoritmos
- Ejemplo: La edad (15-45) y el número de hijos (0-10) quedan en la misma escala

4. Creación de Variables Dummy para Estacionalidad

¿Qué es?

Crear variables "bandera" que indiquen el trimestre.

Ejemplo práctico:

Para el primer trimestre (Ene-Feb-Mar):

```
text
```

```
Q1 = 1, Q2 = 0, Q3 = 0, Q4 = 0
```

Para el segundo trimestre (Abr-May-Jun):

```
text
```

```
Q1 = 0, Q2 = 1, Q3 = 0, Q4 = 0
```

Tabla resultante:

Año	Trimestre	Q1	Q2	Q3	Q4	Nacimientos
2020	1	1	0	0	0	15000
2020	2	0	1	0	0	14500
2020	3	0	0	1	0	14800
2020	4	0	0	0	1	15200

¿Por qué se hace?

- Captura patrones estacionales (ej: más nacimientos en ciertos trimestres)
- Los modelos pueden aprender que el Q4 suele tener más/menos nacimientos

- Ejemplo: En muchos países hay "baby boom" en septiembre (concebidos en navidad)
-



FLUJO COMPLETO DE TRANSFORMACIÓN

Datos Originales → Datos Listos para el Modelo

Paso 1: Agrupar por trimestre

Paso 2: Codificar educación y estado civil

Paso 3: Normalizar edad, número de hijos, etc.

Paso 4: Añadir columnas dummy para trimestres

Paso 5: ¡Listo para entrenar el modelo!

Resultado Final:

Un dataset donde cada fila es un trimestre con:

- Variables numéricas normalizadas (0-1)
 - Variables categóricas codificadas (números)
 - Indicadores de estacionalidad (dummy variables)
 - Número total de nacimientos (variable objetivo)
-



ANALOGÍA SENCILLA

Imagina que estás cocinando:

- Agregación trimestral = Medir ingredientes por tazas, no por gramos individuales
- Codificación = Traducir receta de inglés a español
- Normalización = Usar la misma taza para medir todos los ingredientes

- Variables dummy = Etiquetar ingredientes por temporada (verano/invierno)

¿Te queda claro cada transformación? ¿Quieres que profundice en alguna en particular?

EXPLICACIÓN SENCILLA: VARIABLES DUMMY PARA ESTACIONALIDAD

¿QUÉ SON LAS "VARIABLES DUMMY"?

Imagina que son como interruptores de luz ⚡ para cada trimestre:

- ON (1) = Sí es este trimestre
 - OFF (0) = No es este trimestre
-



EJEMPLO MUY PRÁCTICO

Sin variables dummy:

Año	Trimestre	Nacimientos
2020	1	15,000
2020	2	14,500
2020	3	14,800
2020	4	15,200

Problema: El modelo ve "Trimestre = 2" pero no entiende que es una categoría, piensa que es un número (¡que 2 es el doble que 1!)

CON variables dummy:

Año	Trimestre	Q1	Q2	Q3	Q4	Nacimientos
2020	1	1	0	0	0	15,000
2020	2	0	1	0	0	14,500
2020	3	0	0	1	0	14,800
2020	4	0	0	0	1	15,200

Ahora el modelo entiende:

- "Q1=1, otros=0" → Es el primer trimestre
- "Q2=1, otros=0" → Es el segundo trimestre
- etc.

¿POR QUÉ HACER ESTO?

Para capturar patrones como:

- "En el primer trimestre (ene-mar) siempre nacen más bebés"
- "En el tercer trimestre (jul-sep) hay menos nacimientos"

Ejemplo real:

Si el modelo descubre que:

- Q4 suele tener +5% de nacimientos

- Q2 suele tener -3% de nacimientos

Entonces puede predecir: "Para 2025-Q4, esperamos 5% más nacimientos que el promedio"

CÓMO SE CONSTRUYEN PASO A PASO

Paso 1: Identificar el trimestre de cada fecha

text

Fecha: 15/03/2020 → Trimestre 1 (Ene-Feb-Mar)

Fecha: 20/06/2020 → Trimestre 2 (Abr-May-Jun)

Fecha: 10/09/2020 → Trimestre 3 (Jul-Ago-Sep)

Fecha: 05/12/2020 → Trimestre 4 (Oct-Nov-Dic)

Paso 2: Crear 4 columnas nuevas (Q1, Q2, Q3, Q4)

Paso 3: Poner 1 en la columna correcta

text

Si Trimestre = 1 → Q1=1, Q2=0, Q3=0, Q4=0

Si Trimestre = 2 → Q1=0, Q2=1, Q3=0, Q4=0



Si Trimestre = 3 → Q1=0, Q2=0, Q3=1, Q4=0

Si Trimestre = 4 → Q1=0, Q2=0, Q3=0, Q4=1

ANALOGÍA: CARNÉ DE IDENTIDAD

Piensa en las variables dummy como el carné de identidad de cada trimestre:

Carné del Trimestre 1:

- ¿Eres Q1? Sí  (1)
- ¿Eres Q2? NO  (0)

- ¿Eres Q3? NO ❌ (0)
- ¿Eres Q4? NO ❌ (0)

Carné del Trimestre 2:

- ¿Eres Q1? NO ❌ (0)
 - ¿Eres Q2? SÍ ✅ (1)
 - ¿Eres Q3? NO ❌ (0)
 - ¿Eres Q4? NO ❌ (0)
-



EN TU PROYECTO PERUANO

Patrones que podrías descubrir:

- Q1 (ene-mar): ¿Aumento por "baby boom" de concepción en fiestas patrias?
- Q3 (jul-sep): ¿Disminución por temporada de frío?
- Q4 (oct-dic): ¿Aumento por concepción en meses festivos?

Resultado final:

Tu modelo aprenderá automáticamente:

- "Cuando Q1=1, suma X nacimientos al pronóstico"
 - "Cuando Q3=1, resta Y nacimientos al pronóstico"
-



RESUMEN EN 1 FRASE

"Variables dummy = Crear 4 banderas (Q1,Q2,Q3,Q4) donde solo una está activa por trimestre, para que el modelo aprenda los patrones de cada época del año."

RELACIÓN ENTRE LOS MODELOS: EXPLICACIÓN CLARA

 **SON LOS MISMOS MODELOS, SOLO CAMBIA LA TERMINOLOGÍA**

TABLA DE EQUIVALENCIAS

Modelos que yo mencioné	Modelos que tú mencionas	¿Son iguales?
Regresión Lineal	Regresión lineal	Sí, exactamente el mismo
k-NN Regressor	Regresión KNN	Sí, mismo algoritmo
Decision Trees	Árboles de regresión	Sí, mismo concepto
Random Forest	(No mencionado)	Es un ENSAMBLE de árboles

EXPLICACIÓN DETALLADA POR MODELO

1. REGRESIÓN LINEAL = Regresión lineal

- Exactamente el mismo algoritmo
- Función: Encontrar línea recta que mejor se ajuste a los datos
- Tu proyecto: Predecir tasa natalidad como combinación lineal de variables

2. k-NN REGRESSOR = Regresión KNN

- Exactamente el mismo algoritmo
- Función: Predecir basado en "vecinos más cercanos"
- Ejemplo: Si trimestres similares (mismas características) tuvieron X nacimientos, este trimestre probablemente también

3. DECISION TREES = Árboles de regresión

- Exactamente el mismo concepto
- Función: Crear "reglas de decisión" en forma de árbol
- Ejemplo:
 - ¿Edad madre > 30? → Sí → ¿Nivel educativo > 3? → Sí → Probable 1 hijo

4. RANDOM FOREST

- NO es igual a Árboles de regresión
 - Es un AVANCE: Combina MUCHOS árboles de regresión
 - Ventaja: Más preciso y estable que un solo árbol
-



ANALOGÍA: CONSTRUCCIÓN DE UNA CASA

- Árbol de regresión = Un solo arquitecto haciendo planos
 - Random Forest = Un equipo de 100 arquitectos, cada uno hace planos y votan el mejor diseño
 - Regresión lineal = Usar solo reglas matemáticas simples
 - KNN = Copiar el diseño de casas similares en el vecindario
-



EN TU PROYECTO PRÁCTICO

Flujo recomendado:

1. Empezar con Regresión Lineal (más simple)

2. Probar Árbol de Regresión (captura no linealidades)
3. Usar KNN (para patrones estacionales)
4. FINALMENTE: Random Forest (combinación poderosa de todos los árboles)

Ejemplo de resultados esperados:

text	
Modelo	Precisión (R^2)
Regresión Lineal	0.65
Árbol Regresión	0.72
KNN	0.68
Random Forest	0.85 ← ¡MEJOR!



RESUMEN FINAL

Sí, estás usando los mismos modelos que mencioné, solo que:

- "Decision Trees" = "Árboles de regresión" (mismo)
- "k-NN Regressor" = "Regresión KNN" (mismo)
- "Regresión Lineal" = "Regresión lineal" (mismo)
- Random Forest es la EVOLUCIÓN de los árboles de regresión

¡Tu selección de modelos es perfecta para el proyecto! 🎉

METODOLOGÍA ACTUALIZADA

3.1. Fuente de Datos y Preparación

3.1.1. Dataset Utilizado

Se empleará el dataset "Registros de Nacidos Vivos en el Perú (2015-2025)" del Ministerio de Salud (MINSA) con las siguientes características:

- Período: 2015 a 2025 (datos trimestrales)
- Cobertura: Nacional y desagregada por departamentos
- Variables clave: 22 atributos demográficos y socioculturales

3.1.2. Preprocesamiento de Datos

Siguiendo la metodología CRISP-DM:

Limpieza y Validación:

- Eliminación de registros duplicados e inconsistentes
- Validación de rangos biológicamente posibles:
 - Edad materna: 12-55 años
 - Peso nacido: 500-5,000 gramos
 - Talla nacido: 30-60 cm
- Manejo de valores faltantes mediante interpolación temporal

Selección de Variables:

- `FecNac_Año`, `FecNac_Mes` (series temporales)
- `Edad_Madre`, `Nivel_Instrucción_Madre` (predictores clave)
- `Estado_Civil`, `Hijos_vivo_madre` (factores sociodemográficos)
- `ubiLbigeoInel` (análisis departamental)

3.2. Transformación de Datos

3.2.1. Agregación Trimestral

Conversión de datos individuales a totales por trimestre:

```
python
```

```
# Agrupar por trimestre y departamento
datos_trimestrales = datos.groupby(['Año', 'Trimestre',
'Departamento']).agg({
    'Nacimientos': 'count',
    'Edad_Madre': 'mean',
    'Nivel_Instruccion_Madre': 'mode'
}).reset_index()
```

3.2.2. Codificación de Variables Categóricas

- Nivel educativo: Preescolar=1, Primaria=2, Secundaria=3, Superior=4
- Estado civil: Soltera=1, Conviviente=2, Casada=3, Otros=4

3.2.3. Normalización Min-Max

Escalado de variables numéricas al rango [0,1]:

```
text
```

```
Edad_normalizada = (Edad_actual - 12) / (55 - 12)
```

3.2.4. Variables Dummy para Estacionalidad

Creación de indicadores por trimestre:

Trimestre	Q1	Q2	Q3	Q4
1 (Ene-Mar)	1	0	0	0
2 (Abr-Jun)	0	1	0	0
3 (Jul-Sep)	0	0	1	0
4 (Oct-Dic)	0	0	0	1

3.3. Modelos de Machine Learning Seleccionados

3.3.1. Regresión Lineal Múltiple

Objetivo: Establecer línea base y capturar relaciones lineales

Características:

- Modelo más simple e interpretable
- Ecuación: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$
- Donde Y = Tasa de natalidad, X_i = variables predictoras

Ventajas para tu proyecto:

- Resultados fáciles de explicar
- Bajo costo computacional
- Base para comparar modelos complejos

3.3.2. k-NN Regressor (k-Vecinos Más Cercanos)

Objetivo: Capturar patrones basados en similitud temporal

Características:

- $k = 5$ vecinos (optimizado por validación cruzada)
- Métrica de distancia: Euclidiana
- Busca trimestres históricos con características similares

Aplicación en tu proyecto:

- Si un trimestre tiene características similares a trimestres pasados que tuvieron alta natalidad, predice alta natalidad
- Captura patrones estacionales no lineales

3.3.3. Árboles de Decisión (Decision Trees)

Objetivo: Modelar relaciones no lineales y complejas

Características:

- Profundidad máxima: 10 niveles

- Criterio de división: Reducción de varianza
- Parada temprana para evitar sobreajuste

Ejemplo de reglas que puede aprender:

text

```
SI Edad_Madre > 30  
  Y Nivel_Educativo > 3 (Superior)  
  Y Trimestre = 4
```

ENTONCES → Tasa_Natalidad = X

3.4. Configuración Experimental

3.4.1. División Temporal de Datos

- Entrenamiento: 2015-2019 (5 años - 70%)
- Validación: 2020-2022 (3 años - 20%)
- Testing: 2023-2025 (3 años - 10%)

3.4.2. Métricas de Evaluación

Coeficiente de Determinación (R^2):

- Mide qué porcentaje de la variación es explicado por el modelo
- Rango: 0 a 1 (mejor cerca de 1)

Raíz del Error Cuadrático Medio (RMSE):

- Error promedio en unidades de tasa de natalidad
- Menor valor indica mejor precisión

3.4.3. Estrategia de Validación

- Validación Cruzada Temporal: 5 folds
- Validación de Estabilidad: 3 ejecuciones con diferentes semillas
- Análisis de Residuales: Para detectar patrones no capturados

3.5. Proceso de Entrenamiento y Evaluación

3.5.1. Entrenamiento Individual

Cada modelo se entrena por separado con los mismos datos:

```
python

# Ejemplo de flujo
modelo_lineal.fit(X_entrenamiento, y_entrenamiento)
modelo_knn.fit(X_entrenamiento, y_entrenamiento)

modelo_arbol.fit(X_entrenamiento, y_entrenamiento)
```

3.5.2. Evaluación Comparativa

Tabla de resultados esperada:

Modelo	R ²	RMSE	Ventajas
Regresión Lineal	0.65	0.45	Interpretable
k-NN	0.72	0.38	Captura estacionalidad
Árbol Decisión	0.75	0.35	Captura no linealidades

3.5.3. Análisis de Importancia de Variables

Solo para Árboles de Decisión:

- Método: Importancia por reducción de impureza
- Resultado: Ranking de variables más influyentes
- Aplicación: Identificar factores clave en la natalidad peruana

3.6. Implementación Técnica

3.6.1. Herramientas

- Lenguaje: Python 3.9+
- Librerías: Scikit-learn, Pandas, NumPy, Matplotlib
- Entorno: Google Colab

3.6.2. Flujo de Procesamiento

text

Datos Crudos

↓

Limpieza y Validación

↓

Agregación Trimestral

↓

Transformación (Codificación + Normalización)

↓

Entrenamiento 3 Modelos

↓

Evaluación Comparativa

↓

Análisis de Resultados

3.7. Ventajas de Esta Selección de Modelos

3.7.1. Diversidad de Enfoques

- Lineal: Relaciones simples y globales
- k-NN: Patrones locales y de similitud
- Árbol: Relaciones complejas y no lineales

3.7.2. Balance Complejidad-Interpretabilidad

- Todos los modelos son interpretables
- Permiten entender qué factores afectan la natalidad
- Resultados comunicables a no técnicos

3.7.3. Adecuación al Problema

- Capturan tanto tendencias temporales como factores sociodemográficos
- Apropriados para el volumen de datos disponible
- Computacionalmente eficientes