Proyecto Final Minería de Datos
Documentación de la Metodología CRISP- DM
Predicción de la Clasificación Final del Dengue a partir de datos de vigilancia epidemiológica en Colombia

Stefany Morelos Morelo
Daniel Elías Córdoba Howard
Daniela Gerena Lopera

Universidad Pontificia Bolivariana
Ingeniería en ciencia de datos
Analítica de datos estructurados (Minería de datos)
Profesora Ana Isabel Oviedo Carrascal

Medellín, Colombia
Mayo de 2025

Tabla de contenido

Lista de tablas

1. Entendimiento del negocio

1.1. Descripción del negocio

El dengue es una enfermedad viral transmitida por mosquitos que representa una carga significativa para el sistema de salud pública. El Instituto Nacional de Salud (INS) recopila y publica datos detallados sobre los casos reportados en la ciudad de Bucaramanga, incluyendo la clasificación final de la enfermedad en función de la severidad de los síntomas.

1.2. Descripción del problema

La identificación oportuna y precisa de los casos de dengue según su gravedad (sin signos de alarma, con signos de alarma o dengue grave) es esencial para asignar recursos médicos, prevenir complicaciones y reducir la mortalidad. Actualmente, esta clasificación depende del análisis clínico posterior a la notificación, lo cual puede retrasar la atención adecuada.

1.3. Objetivos de la minería

Desarrollar un modelo predictivo basado en técnicas de minería de datos que realice la predicción la clasificación final del dengue.

1.4. Diseño de la solución

Tabla 1. Diseño de la solución

Descripción del Problema	Tipo de análisis (Predictivo/Descriptivo)	Tipo de aprendizaje	Tarea analítica	Requerimiento en los datos	Métodos	Evaluación
Determinar cuál va a ser la clasificación final de dengue en Bucaramanga	Predictivo	Supervisado	Clasificación	Debe haber histórico de datos Debe haber una sola variable objetivo categórica Debe haber relación entre predictoras y objetivo No debería existir relación entre variables predictoras	Redes neuronales SVM Árboles de decisión Knn	Matriz de confusión Área ROC F1 score Precision Exactitud Recall
Determinar cuál va a ser la clasificación final de dengue en Bucaramanga	Predictivo	De ensambles	Clasificación	Debe haber histórico de datos Debe haber una sola variable objetivo categórica Debe haber relación entre predictoras y objetivo No debería existir relación entre variables predictoras	Bagging XGBoost Stacking	Matriz de confusión Área ROC F1 score Precision Exactitud Recall

1.5. Recursos para la creación del modelo y para despliegue

Hardware utilizado

- **Entorno local y en la nube**: El desarrollo y entrenamiento del modelo se realizó en **Google Colab**, que ofrece recursos en la nube sin necesidad de hardware local potente.
- **Recursos de Colab**:
  - CPU compartida
  - Memoria RAM de hasta 12 GB
  - Almacenamiento temporal de archivos `.pkl` y `.xlsx`

Software y librerías

- **Lenguaje de programación**: Python 3.10+
- **Librerías principales**:
  - `scikit-learn`: para los modelos de ML tradicionales (KNN, SVM, árboles, etc.)
  - `xgboost`: para el modelo de boosting
  - `imblearn`: para el balanceo con SMOTEN
  - `pandas`, `numpy`, `matplotlib`, `seaborn`: para análisis de datos y visualización
  - `cloudpickle` / `pickle`: para guardar y cargar el modelo
  - `streamlit`: para el despliegue de la aplicación web
  - `scikit-optimize`: para optimización bayesiana de hiperparámetros

2. Entendimiento de los datos

2.1. Ciclo de los datos

El conjunto de datos corresponde a registros de casos de dengue reportados en el municipio de Bucaramanga. La información abarca variables clínicas, demográficas, socioeconómicas y geográficas de los pacientes. Estos datos hacen parte de un sistema de vigilancia epidemiológica y son recolectados de manera continua por las entidades de salud.

El ciclo de vida de estos datos puede resumirse así:

- **Recolección:** por parte de centros médicos, hospitales y EPS, quienes notifican casos de dengue al sistema de salud. Para este caso específico, se usarán datos recopilados por el Instituto Nacional de Salud (INS).
- **Almacenamiento:** en bases de datos del sistema de vigilancia, con estructuras tabulares.
- **Procesamiento:** mediante limpieza, validación y clasificación de los registros.
- **Análisis:** para identificar patrones, focos de brote, factores de riesgo y comportamientos temporales de la enfermedad.
- **Toma de decisiones:** los resultados se utilizan para guiar campañas de prevención, intervenciones sanitarias y asignación de recursos.

2.2. Diccionario de datos

1. **orden:** Consecutivo autonumérico.
2. **cod_eve:** Código del evento de interés en salud pública.
3. **grupo:** Nombre del grupo de interés en salud pública.
4. **fec_not:** Fecha de notificación del evento de interés.
5. **semana:** Semana Epidemiológica: 1-52 del evento de interés.
6. **Año:** Año de la Semana Epidemiológica.
7. **grupo_etario:** Clasificación por edad y ciclo vital humano (ej: menor que 1; 1-4; 5-9; etc.).
8. **clasif_edad:** Rango de edades para clasificación de ciclos de vida.
9. **def_clas_edad:** Clasificación de la edad según el Ministerio de Salud (ej: Primera infancia, Adolescencia, etc.).
10. **sexo_:** Característica fisiológica al nacer (M = Masculino, F = Femenino, I = Indeterminado).
11. **ocupacion_:** Código de clasificación de actividad comercial.

12. **tip_ss_:** Tipo de régimen de seguridad social (ej: Contributivo, Subsidiado, etc.).
13. **cod_ase_:** Código de la aseguradora o EPS.
14. **aseguradora:** Nombre de la aseguradora.
15. **per_etn_:** Pertenencia étnica (ej: Indígena, Raizal, Afrocolombiano, etc.).
16. **estrato_:** Clasificación socioeconómica del inmueble de residencia.
17. **gp_discapa:** Población con discapacidad (1 = Sí, 2 = No).
18. **gp_desplaz:** Víctima de desplazamiento forzado (1 = Sí, 2 = No).
19. **gp_migrant:** Población migrante (1 = Sí, 2 = No).
20. **gp_carcela:** Población carcelaria (1 = Sí, 2 = No).
21. **gp_gestan:** Población gestante (1 = Sí, 2 = No).
22. **sem_ges_:** Número de la semana gestacional.
23. **gp_indigen:** Población indígena (1 = Sí, 2 = No).
24. **gp_pobicbf:** Población vinculada al ICBF (1 = Sí, 2 = No).
25. **gp_mad_com:** Madres comunitarias (1 = Sí, 2 = No).
26. **gp_desmovi:** Población desmovilizada de grupos armados (1 = Sí, 2 = No).
27. **gp_psiquia:** Población psiquiátrica (1 = Sí, 2 = No).
28. **gp_vic_vio:** Víctima de violencia (1 = Sí, 2 = No).
29. **gp_otros:** Otras poblaciones (1 = Sí, 2 = No).
30. **fuente_:** Fuente de notificación (ej: Rutinaria, Búsqueda activa, etc.).
31. **cod_pais_r:** Código DANE del país de residencia.
32. **cod_dpto_r:** Código DANE del departamento de residencia.
33. **cod_mun_r:** Código DANE del municipio de residencia.
34. **fec_con_:** Fecha de consulta a servicio médico.
35. **ini_sin_:** Fecha de inicio de síntomas.
36. **tip_cas_:** Tipo de caso (ej: Sospechoso, Confirmado por laboratorio, etc.).
37. **pac_hos_:** Paciente hospitalizado (1 = Sí, 2 = No).
38. **fec_hos_:** Fecha de hospitalización.
39. **con_fin_:** Estado del usuario (Vivo, Muerto, No sabe).
40. **fec_def_:** Fecha de defunción.
41. **ajuste_:** Estadio del caso (ej: Sin ajuste, Descartado, etc.).
42. **versión:** Versión del sistema de vigilancia epidemiológica.
43. **desplazami:** Desplazamiento en los últimos 15 días (1 = Sí, 2 = No).
44. **famantdngu:** Familiar con síntomas de dengue (1 = Sí, 2 = No, 3 = Desconocido).

45. **fiebre:** Fiebre en dengue sin signos de alarma (1 = Sí, 2 = No).
46. **cefalea:** Cefalea en dengue sin signos de alarma (1 = Sí, 2 = No).
47. **dolrretroo:** Dolor retroocular en dengue sin signos de alarma (1 = Sí, 2 = No).
48. **malgias:** Mialgias en dengue sin signos de alarma (1 = Sí, 2 = No).
49. **artralgia:** Artralgias en dengue sin signos de alarma (1 = Sí, 2 = No).
50. **erupcionr:** Erupción cutánea en dengue sin signos de alarma (1 = Sí, 2 = No).
51. **dolor_abdo:** Dolor abdominal en dengue con signos de alarma (1 = Sí, 2 = No).
52. **vomito:** Vómito en dengue con signos de alarma (1 = Sí, 2 = No).
53. **diarrea:** Diarrea en dengue con signos de alarma (1 = Sí, 2 = No).
54. **somnolenci:** Somnolencia o irritabilidad en dengue con signos de alarma (1 = Sí, 2 = No).
55. **hipotensio:** Hipotensión en dengue con signos de alarma (1 = Sí, 2 = No).
56. **hepatomeg:** Hepatomegalia en dengue con signos de alarma (1 = Sí, 2 = No).
57. **hem_mucosa:** Hemorragias en mucosas en dengue con signos de alarma (1 = Sí, 2 = No).
58. **hipotermia**: Hipotermia en dengue con signos de alarma (1 = Sí, 2 = No).
59. **aum_hemato:** Aumento de hematocrito en dengue con signos de alarma (1 = Sí, 2 = No).
60. **caida_plaq:** Plaquetas <100.000 en dengue con signos de alarma (1 = Sí, 2 = No).
61. **acum_liqui:** Acumulación de líquidos en dengue con signos de alarma (1 = Sí, 2 = No).
62. **extravasac:** Extravasación severa de plasma en dengue grave (1 = Sí, 2 = No).
63. **hemorr_hem:** Hemorragia hemodinámicamente significativa en dengue grave (1 = Sí, 2 = No).
64. **choque**: Shock por dengue grave (1 = Sí, 2 = No).
65. **daño_organ:** Daño grave de órganos en dengue grave (1 = Sí, 2 = No).
66. **muesttejid:** Muestra en tejido en mortalidad por dengue (1 = Sí, 2 = No).
67. **mueshigado:** Muestra en hígado en mortalidad por dengue (1 = Sí, 2 = No).
68. **muesbazo:** Muestra en bazo en mortalidad por dengue (1 = Sí, 2 = No).
69. **muespulmon:** Muestra en pulmón en mortalidad por dengue (1 = Sí, 2 = No).
70. **muescerebr:** Muestra en cerebro en mortalidad por dengue (1 = Sí, 2 = No).
71. **muesmiocar:** Muestra en miocardio en mortalidad por dengue (1 = Sí, 2 = No).

72. **muesmedula:** Muestra en médula en mortalidad por dengue (1 = Sí, 2 = No).
73. **muesriñon:** Muestra en riñón en mortalidad por dengue (1 = Sí, 2 = No).
74. **clasfinal:** Clasificación final del caso (ej: Dengue sin signos, Dengue grave).
75. **conducta:** Conducta médica (ej: Ambulatoria, Hospitalización, UCI).
76. **nom_eve:** Nombre del evento de interés en salud pública.
77. **COMUNA shp:** Comuna del municipio de Bucaramanga (ej: Norte, Sur, Centro).
78. ***BARRIO_VER* shp:** Barrio dentro de la comuna (ej: Lagos del Cacique, Provenza).

Notas:

- Algunos campos tienen descripciones genéricas o valores predeterminados como "No sabe" debido a limitaciones en los metadatos originales.
- Los tipos de datos y longitudes se omitieron para simplificar el listado, pero están disponibles en la ficha técnica.

2.3. Reglas de calidad

Durante la exploración inicial, se identificaron las siguientes reglas y aspectos de calidad relevantes:

- **Valores faltantes**: la variable `ocupacion_` presenta valores nulos. Se evaluará su tratamiento durante la fase de limpieza.
- **Codificación binaria inconsistente**: los síntomas están codificados como 1 (Sí), 2 (No), pero en algunos contextos puede considerarse usar 0 y 1 para análisis más consistentes.
- **Variables con formato fecha**: varias variables como `fec_not`, `ini_sin_`, `fec_hos_`, etc., están en formato string y deben ser transformadas a tipo datetime.
- **Duplicados**: no se identificaron registros duplicados al primer análisis, pero se verificará más adelante.
- **Consistencia de variables categóricas**: se validará que las categorías como `sexo_`, `grupo_etario`, `clasfinal`, etc., no presenten errores de digitación o codificación fuera del dominio esperado.

- Columnas como `estrato_`, `sem_ges_`, y `cod_pais_r` contienen "-89", que probablemente indica datos no registrados. Estos se reemplazarán con NA.

3. Preparación de datos

3.1. Integración

En nuestro caso de estudio, no fue necesaria realizar una integración de diversas fuentes para tener los datos necesarios para la creación de los modelos. Por ende, provienen de:

Entidad responsable: Secretaría de Salud y Ambiente Municipal de Bucaramanga.
Licencia: Creative Commons Attribution Share Alike 4.0 International.
Enlace: Datos Abiertos Bucaramanga.
Frecuencia de Actualización: Mensual.

3.2. Selección de variables

Descripción:
Se realizó una depuración de variables eliminando aquellas que eran irrelevantes para el análisis, como identificadores (`orden`) y fechas redundantes (`semana`, `fec_not`, etc.). Esto se justificó por su nula o escasa capacidad para aportar información útil al modelo predictivo.

```
df = df.drop('orden',axis=1)

df = df.drop('semana',axis=1)

df = df.drop('fec_not',axis=1)

df = df.drop('fec_con_',axis=1)

df = df.drop('fec_hos_',axis=1)

df = df.drop('fec_def_',axis=1)
```

3.3. Descripción estadística

Descripción:

Se generó un resumen estadístico del conjunto de datos mediante el método. describe(), y adicionalmente, se utilizó la librería `pandas_profiling` (ahora `ydata-profiling`) para obtener un perfil completo del dataset.

```
df.describe()
from pandas_profiling import ProfileReport
profile_df=ProfileReport(df, minimal=False)
```

3.4. Limpieza de atípicos

Descripción:

Se identificaron variables con valores atípicos evidentes o codificados erróneamente (como `-89`), lo cual indicaba errores de digitación o datos faltantes mal codificados. Estas variables fueron eliminadas si más del 20% de sus registros eran inválidos.

```
df = df.drop(columns=['estrato_', 'sem_ges_', 'fuente_', 'cod_pais_r', 'famantdngu'], axis=1)
```

3.5. Limpieza de nulos

Descripción:

Se realizó imputación de valores faltantes usando la moda en variables categóricas con menos del 15% de nulos, como `ocupacion_`, utilizando `SimpleImputer` de `sklearn`.

```
from sklearn.impute import SimpleImputer ImpCategorias = SimpleImputer(missing_values=np.nan, strategy='most_frequent') df['ocupacion_'] = ImpCategorias.fit_transform(df[['ocupacion_']])
```

3.6. Creación de nuevas variables

Se transformó la variable `ini_sin_` (fecha de inicio de síntomas) extrayendo solo el **mes** como una nueva variable categórica. También se limpió y transformó la variable `version` para quedarse únicamente con el año.

df['version'] = df['version'].str.extract(r'(\d{4})').astype('category') df = reemplazar_por_mes(df, 'ini_sin_') df['ini_sin_'] = df['ini_sin_'].astype('category')

3.7. Análisis de correlaciones para redundancia

Descripción:
Con el fin de evitar multicolinealidad entre variables predictoras, se calculó una matriz de correlación sobre el conjunto de datos codificado con variables dummy. Se identificaron pares de variables altamente correlacionadas entre sí (correlación absoluta mayor a 0.8). Para cada par, se eliminó la variable con menor correlación frente a la variable objetivo (`clasfinal`), con el fin de conservar la de mayor aporte predictivo.

umbral_alta_corr = 0.8 predictores_a_eliminar = set()

for i in range(len(corr_matrix.columns)): for j in range(i + 1, len(corr_matrix.columns)): predictor_i = corr_matrix.columns[i] predictor_j = corr_matrix.columns[j] if predictor_i != 'clasfinal' and predictor_j != 'clasfinal': corr_ij = abs(corr_matrix.iloc[i, j]) if corr_ij >= umbral_alta_corr: if abs(objetivo_corr[predictor_i]) > abs(objetivo_corr[predictor_j]): predictores_a_eliminar.add(predictor_j) else: predictores_a_eliminar.add(predictor_i)

3.8. Análisis de correlaciones para irrelevancia (predicciones)

Descripción:
Se analizó la fuerza de asociación lineal entre cada variable predictora y la variable objetivo (`clasfinal`), previamente transformada a una escala ordinal (0, 1, 2) para representar sus categorías. Aquellas variables cuya correlación con el objetivo fue muy baja (valor absoluto menor a 0.1) fueron consideradas irrelevantes para el modelo de clasificación y se eliminaron del conjunto de datos.

Esta estrategia se utiliza como una heurística preliminar de selección de características en clasificación multiclase.

umbral_baja_corr = 0.1 predictores_baja_corr = objetivo_corr[abs(objetivo_corr) < umbral_baja_corr].index.tolist()

3.9. Reducción de dimensión (opcional en predicciones)

Descripción:

La reducción de dimensión se realizó mediante la combinación de los dos análisis de correlación anteriores: eliminación de variables altamente redundantes y de baja relevancia. No se utilizó PCA ni técnicas avanzadas, ya que la reducción fue manual y dirigida al problema de clasificación.

3.10. Balanceo (clasificación)

Descripción:

Dado que la distribución original de clases era desequilibrada, se aplicó la técnica **SMOTEN (Synthetic Minority Over-sampling Technique for Nominal data)** al conjunto de entrenamiento (70%) para balancear la clase minoritaria ("Dengue grave") hasta alcanzar el 50% del tamaño de la clase mayoritaria. Esto permitió mitigar el sesgo hacia la clase dominante y mejorar la capacidad de los modelos para detectar casos graves.

3.11. Transformaciones

Descripción:

Después del balanceo, se aplicó **codificación one-hot (dummies)** a todas las variables categóricas, ya que los algoritmos de clasificación requieren entradas numéricas. Además, se evaluó si era necesario aplicar normalización, pero se concluyó que no había variables numéricas continuas, por lo que esta transformación fue innecesaria.

4. Modelamiento, evaluación e interpretación

Descripción:

Se entrenaron múltiples modelos de clasificación con validación cruzada estratificada de 10 particiones sobre el conjunto de entrenamiento balanceado. Las métricas de evaluación utilizadas fueron: **F1-score ponderado**, **accuracy**, **recall**

macro y **AUC multicategoría (ROC_AUC_OVO)**.

Posteriormente, los modelos se evaluaron con el conjunto de prueba (30%) **no balanceado**, para obtener métricas en condiciones reales.

Modelos evaluados:

- Árbol de Decisión
- K-Nearest Neighbors (KNN)
- Red Neuronal (MLP)
- SVM
- Bagging con KNN
- XGBoost

4.1. Configuración métodos de machine learning

Para abordar el problema de clasificación de casos de dengue, se configuraron y evaluaron diversos modelos de aprendizaje supervisado. El conjunto de entrenamiento fue previamente balanceado utilizando **SMOTEN**, dado que las variables eran categóricas.

Se utilizaron los siguientes algoritmos:

- **Árbol de Decisión** (`DecisionTreeClassifier`): sin normalización, debido a que los árboles no se ven afectados por la escala de los datos.
- **K-Nearest Neighbors (KNN)**: configurado con métrica `hamming`, adecuada para datos categóricos.
- **Red Neuronal** (`MLPClassifier`): activación ReLU, `early_stopping`, `solver='adam'`.
- **SVM** (`SVC`): con `kernel` lineal y probabilidades activadas para el cálculo del AUC.
- **Bagging**: usando KNN como estimador base.
- **XGBoost**: con configuración optimizada y codificación de clases ya preprocesada.
- **Stacking**: combinación de los modelos base con regresión logística como modelo ensamblador.

Cada modelo fue evaluado mediante **validación cruzada estratificada (10 folds)** utilizando las siguientes métricas:

- `f1_weighted`
- `accuracy`
- `recall_macro`
- `roc_auc_ovo`

4.2. Análisis de medidas de calidad

Se creó un `DataFrame` llamado `comparacion_CV` para almacenar los resultados de validación cruzada de cada modelo. Las métricas principales comparadas fueron el **F1-score ponderado (f1_weighted)** en entrenamiento y prueba.

Para comparar formalmente los modelos se usó:

- **ANOVA** para identificar si existían diferencias estadísticas significativas entre los modelos.
- En caso de diferencias, se aplicó **Tukey HSD** para identificar pares de modelos con diferencias significativas.
- Se utilizaron gráficos tipo **boxplot** para visualizar la distribución de rendimiento por modelo.

4.3. Selección del mejor modelo

Tras el análisis estadístico y comparativo, se identificaron los **tres modelos con mejor rendimiento** y menor complejidad computacional:

- **Red Neuronal (RN)**
- **SVM**
- **KNN**

Estos modelos fueron seleccionados debido a que:

- Presentaron los **mejores F1-score en el conjunto de prueba real (no balanceado)**.
- No mostraron diferencias estadísticas significativas entre sí según Tukey.
- Tienen **menor costo computacional** que los modelos de ensamble (como Stacking o XGBoost).

A estos tres modelos se les aplicó posteriormente:

- **Hiperparametrización con GridSearchCV**
- **Optimización Bayesiana con BayesSearchCV**

Finalmente, se compararon los tres modelos optimizados, y el mejor se seleccionó según el **F1-score en el conjunto de prueba**. Aunque el código no guarda explícitamente el "ganador absoluto", el modelo con **mayor F1-weighted en `resultados_test`** representa el **modelo final del proyecto.**

5. Despliegue

El modelo seleccionado fue entrenado completamente con los datos balanceados (70% del total) y probado sobre los datos reales (30%). Para el despliegue se recomienda:

- **Guardar el modelo entrenado** con librerías como `joblib` o `pickle`.
- Exportar también el pipeline de preprocesamiento (conversión a dummies, label encoder, etc.).

- Construir una función o servicio API donde se reciban los nuevos registros de pacientes y se obtenga la predicción de clasificación (nivel de gravedad del dengue). En nuestro caso Streamlit.

5.1. Predicción de datos futuros

Entorno de prueba y despliegue

- El modelo está siendo desplegado inicialmente en **Google Colab** como entorno de pruebas, donde se puede cargar el modelo entrenado (`.pkl`) y realizar predicciones sobre nuevos datos manualmente.
- Posteriormente, el objetivo es llevar el modelo a una **aplicación web interactiva con Streamlit**, accesible desde cualquier navegador web.

Tipo de dispositivo y plataforma

- **Dispositivo de uso final**: Computadora de escritorio, portátil, tablet o incluso dispositivo móvil.
- **Plataforma de acceso**: Aplicación web accesible vía navegador.
- **Requisitos del usuario**:
  - Navegador web moderno (Chrome, Firefox, Edge)
  - Acceso a Internet

5.2. Monitoreo

Se recomienda implementar un sistema de monitoreo que permita registrar métricas clave (por ejemplo, F1-score mensual en producción) y detectar posibles degradaciones de desempeño, lo cual es común ante cambios en la distribución de los datos (drift).

5.3. Cronograma de mantenimiento/Reentrenamiento

Etapa	Frecuencia sugerida	Actividad clave
Evaluación del desempeño en producción	Mensual	Comparar predicciones vs. realidad
Validación de integridad de datos nuevos	Semanal	Verificar estructuras y valores esperados
Reentrenamiento del modelo	Trimestral	Incorporar nuevos datos y ajustar el modelo
Reoptimización de hiperparámetros	Semestral	Aplicar Grid o Bayesian Optimization
Auditoría general del sistema	Anual	Revisar pipeline completo de ML