

**CASO PRÁCTICTO PROPUESTO IDL3**

**MACHINE LEARNING I**

**Elaborado por:**

**NEK RYAN JIMENEZ MANDUJANO**

**Solicitado por:**

**SERGIO VICTOR ORIZANO SALVADOR**

**Huancayo, 2025**

**IDL3 – Machine Learning I**

**Modelo Predictivo de Cambio de Hábitos por Temor a la Inseguridad en Lima Metropolitana (ENAPRES 2023 – Capítulo 600)**

**Introducción**

El presente trabajo tiene como objetivo el desarrollo de un **modelo predictivo de Machine Learning** que permita identificar el **cambio de hábitos por temor a la inseguridad ciudadana en Lima Metropolitana**, a partir de los datos de la **Encuesta Nacional de Programas Presupuestales (ENAPRES 2023, Capítulo 600)**.

Se implementaron diferentes fases del ciclo de aprendizaje automático: preparación y exploración de los datos, entrenamiento de modelos base, validación cruzada, optimización de hiperparámetros, análisis de sesgo y varianza, y generación de visualizaciones.

Este estudio busca no solo evaluar el desempeño de distintos algoritmos de clasificación (Árbol de Decisión, Random Forest, XGBoost y Regresión Logística), sino también aportar evidencia útil para la formulación de estrategias de seguridad ciudadana basadas en datos.

**Enlace al repositorio GitHub**

En el siguiente repositorio se encuentra el **código completo del proyecto**, junto con los archivos de resultados exportados y las visualizaciones generadas:

**Repositorio GitHub:**

**1. Preparación del entorno y datos**

**1.1 Importación de librerías**

Para el desarrollo del proyecto se emplearon las siguientes librerías de Python:

* **pandas** y **numpy**: manipulación, limpieza y análisis de datos.
* **scikit-learn**: construcción, entrenamiento y evaluación de modelos de Machine Learning.
* **matplotlib** y **seaborn**: visualización y análisis gráfico de variables.
* **xgboost**: implementación de modelos de ensamble con gradiente boosting de alto rendimiento.

Estas herramientas permiten realizar todo el ciclo de procesamiento, desde la preparación del dataset hasta la generación de visualizaciones y la evaluación de modelos predictivos.

**1.2 Carga del dataset**

El dataset utilizado corresponde al archivo **CAP600\_data\_limpia.csv**, proveniente de la **Encuesta Nacional de Programas Presupuestales (ENAPRES) 2023**, filtrado para el ámbito de **Lima Metropolitana**.  
La base contiene información sociodemográfica, percepción de inseguridad ciudadana, experiencias de victimización y cambios de hábitos frente al temor a la delincuencia.

**1.3 Exploración inicial**

En la exploración preliminar se obtuvieron los siguientes hallazgos:

* **Número de registros:** 7,767 observaciones válidas.
* **Número de variables predictoras utilizadas:** 113 (seleccionadas de un set inicial de más de 200 columnas originales, tras procesos de limpieza y renombrado).
* **Tipos de datos:** mezcla de variables categóricas (e.g., sexo, estrato, percepción de inseguridad) y numéricas (e.g., edad).
* **Valores nulos:** presencia mínima, tratados mediante imputación de mediana para variables numéricas y moda para variables categóricas.

**1.4 Visualización inicial**

La variable objetivo **CAMBIO\_HABITO** es binaria:

* Valor **0**: ciudadanos que no cambiaron sus hábitos por temor a la inseguridad.
* Valor **1**: ciudadanos que sí modificaron al menos un hábito cotidiano.

De acuerdo con la proporción de clases obtenida:

* **61.3% de los encuestados reporta cambios de hábitos.**
* **38.7% no modificó sus hábitos.**

Esto indica que existe una prevalencia considerable de ciudadanos que ajustaron su comportamiento cotidiano frente al temor a la delincuencia, lo que refuerza la relevancia del análisis predictivo.

Adicionalmente, se generaron histogramas por **edad, estrato y sexo**, observándose diferencias en la distribución de la variable objetivo según estas características.

**2. Entrenamiento de un modelo básico**

**2.1 Selección de features**

Se seleccionaron las siguientes variables predictoras:

* **Sociodemográficas:** sexo, edad, estrato.
* **Victimización:** delitos sufridos (P601).
* **Denuncia y frecuencia:** variables relacionadas a frecuencia de victimización (P602), denuncia (P606) y resultados de la denuncia (P608).
* **Percepción:** inseguridad en el barrio (P611\_1).
* **Confianza institucional:** confianza en la policía (P611D\_1).

La variable objetivo fue **CAMBIO\_HABITO**, que identifica si el ciudadano modificó sus hábitos por temor a la inseguridad.

**2.2 División de datos**

Se aplicó la técnica de **Hold-Out**, dividiendo la muestra en un **70% para entrenamiento** y **30% para prueba**, con estratificación respecto a la variable objetivo para mantener la proporción de clases.

**2.3 Modelos iniciales**

Como modelos base se entrenaron:

* **Árbol de Decisión**.
* **Random Forest**.

Estos algoritmos permiten establecer un punto de referencia para la comparación posterior con modelos optimizados.

**2.4 Evaluación inicial**

La evaluación se realizó con métricas de clasificación: **Accuracy, Recall, F1-score, ROC-AUC** y matriz de confusión.

**Árbol de Decisión**

* **Accuracy:** 57.2%
* **Recall (clase 1 – cambio de hábito):** 0.606
* **F1-score (clase 1):** 0.634
* **ROC-AUC:** 0.583
* **Matriz de confusión:**
  + Verdaderos negativos (0 clasificado correctamente): 467
  + Falsos positivos: 436
  + Falsos negativos: 562
  + Verdaderos positivos: 866

Interpretación: el Árbol de Decisión muestra un desempeño moderado, con recall aceptable pero con un número considerable de falsos negativos.

**Random Forest**

* **Accuracy:** 60.8%
* **Recall (clase 1 – cambio de hábito):** 0.721
* **F1-score (clase 1):** 0.692
* **ROC-AUC:** 0.611
* **Matriz de confusión:**
  + Verdaderos negativos: 388
  + Falsos positivos: 515
  + Falsos negativos: 399
  + Verdaderos positivos: 1029

Interpretación: el Random Forest supera al Árbol de Decisión en todas las métricas, mostrando mejor capacidad para detectar ciudadanos que cambiaron hábitos (mayor recall), aunque aún se observan errores de clasificación en la clase 0 (sin cambios).

**Conclusión de la fase**

El **Random Forest** se posiciona como el mejor modelo base, con métricas superiores y un equilibrio más favorable entre recall y F1-score. Estos resultados servirán como referencia para la validación cruzada y posterior optimización de hiperparámetros.

**3. Validación cruzada**

**3.1 Técnica utilizada**

Para evaluar la estabilidad y capacidad de generalización de los modelos, se aplicó **K-Fold Cross Validation con k=5**. Esta técnica divide el dataset en cinco subconjuntos, entrenando en cuatro y validando en el restante de manera iterativa, de modo que todos los registros participan en entrenamiento y validación.

Se aplicó la validación cruzada a los siguientes algoritmos:

* Regresión Logística
* Árbol de Decisión
* Random Forest
* XGBoost

**3.2 Resultados**

Los resultados promedio de las métricas se presentan en la siguiente tabla:

| **Modelo** | **Accuracy** | **Precision** | **Recall** | **F1** | **ROC-AUC** |
| --- | --- | --- | --- | --- | --- |
| Regresión Logística | 0.604 | 0.779 | 0.494 | 0.604 | 0.679 |
| Árbol de Decisión | 0.584 | 0.681 | 0.604 | 0.640 | 0.603 |
| Random Forest | 0.618 | 0.696 | 0.669 | 0.682 | 0.644 |
| XGBoost | **0.642** | 0.687 | **0.764** | **0.723** | **0.689** |

**Interpretación:**

* **XGBoost** obtuvo los mejores resultados globales, con un **Recall de 0.764**, **F1 de 0.723** y el mayor ROC-AUC (0.689). Esto indica una fuerte capacidad para identificar a los ciudadanos que cambiaron hábitos.
* **Random Forest** se posicionó como el segundo mejor, con métricas competitivas, especialmente en F1 (0.682).
* **Regresión Logística** mostró resultados más modestos, con buen nivel de precisión (0.779), pero menor Recall, lo que implica que clasifica menos ciudadanos de la clase positiva (cambio de hábitos).
* **Árbol de Decisión** evidenció desempeño inestable, con menor capacidad de generalización, mostrando una tendencia al sobreajuste.

**3.3 Conclusión**

La validación cruzada permitió confirmar que los **modelos de ensamble (Random Forest y XGBoost)** son los más adecuados para este problema, combinando mejor estabilidad, Recall y F1-score.  
En particular, **XGBoost** sobresale como el modelo con mayor capacidad de generalización, siendo un fuerte candidato para la etapa de optimización de hiperparámetros.

**4. Optimización de hiperparámetros**

**4.1 Técnicas aplicadas**

Para mejorar el rendimiento de los modelos seleccionados, se aplicaron dos enfoques de búsqueda de hiperparámetros:

* **Grid Search:** exploración exhaustiva de combinaciones de parámetros predefinidos.
* **Random Search:** búsqueda aleatoria dentro de un espacio de parámetros, lo que reduce significativamente el costo computacional.

**4.2 Parámetros ajustados**

En los modelos se ajustaron los siguientes hiperparámetros críticos:

* **Random Forest:**
  + Número de árboles (*n\_estimators*).
  + Profundidad máxima (*max\_depth*).
  + Número mínimo de muestras para división (*min\_samples\_split*).
  + Número máximo de características a considerar en cada división (*max\_features*).
* **XGBoost:**
  + Número de árboles (*n\_estimators*).
  + Profundidad máxima (*max\_depth*).
  + Tasa de aprendizaje (*learning\_rate*).
  + Submuestreo de instancias (*subsample*).
  + Submuestreo de características (*colsample\_bytree*).

**4.3 Resultados de la búsqueda**

Los mejores hiperparámetros encontrados fueron:

* **Random Forest (versión ligera):**

{'clf\_\_max\_depth': 10, 'clf\_\_min\_samples\_split': 2, 'clf\_\_n\_estimators': 100}

* **XGBoost (versión ligera):**

{'clf\_\_subsample': 1, 'clf\_\_n\_estimators': 100, 'clf\_\_max\_depth': 3, 'clf\_\_learning\_rate': 0.05, 'clf\_\_colsample\_bytree': 1}

**Nota**

Dado que el dataset cuenta con **7,767 registros y más de 100 variables predictoras**, la búsqueda exhaustiva de hiperparámetros hubiera requerido un tiempo de cómputo elevado.  
Por ello, en este trabajo académico se aplicó una **versión reducida de la optimización**, limitando el número de combinaciones evaluadas en *Grid Search* y el número de iteraciones en *Random Search*.

Esto permitió obtener resultados representativos de la metodología sin comprometer la viabilidad del proyecto. En un entorno productivo, se recomienda ampliar los rangos de búsqueda y aplicar técnicas más avanzadas (como *Bayesian Optimization* o *Hyperopt*) para lograr un ajuste más fino de los modelos.

**4.4 Selección del modelo optimizado**

* El **Random Forest optimizado** mostró un balance adecuado entre **Recall y F1-score**, ofreciendo un desempeño robusto.
* El **XGBoost optimizado** alcanzó el **mejor Recall**, lo que lo hace especialmente útil para identificar a los ciudadanos que cambiaron hábitos, aunque con un costo en interpretabilidad frente a Random Forest.

**5. Evaluación del modelo optimizado**

**5.1 Métricas recalculadas**

Tras aplicar los hiperparámetros optimizados en **Random Forest** y **XGBoost**, se obtuvo una mejora respecto a los modelos base:

* **Random Forest optimizado**:
  + Accuracy: 64%
  + Recall (clase positiva – cambio de hábito): 0.94
  + F1-score (clase positiva): 0.76
  + ROC-AUC: 0.682
* **XGBoost optimizado**:
  + Accuracy: 64%
  + Recall (clase positiva – cambio de hábito): 0.89
  + F1-score (clase positiva): 0.75
  + ROC-AUC: 0.684

En ambos modelos se alcanzó un **F1-score superior a 0.70** en la clase positiva, con un **Recall elevado**, lo que implica que los modelos optimizados mejoran sustancialmente la capacidad de identificar a los ciudadanos que modificaron sus hábitos por temor a la delincuencia.

**5.2 Visualizaciones comparativas**

* **Curvas ROC y Precision-Recall:** al comparar los modelos básicos con los optimizados, se observa un incremento en el área bajo la curva, confirmando la mejora en la capacidad discriminativa.
* **Matrices de confusión:** muestran una **reducción importante de falsos negativos** en los modelos optimizados, lo que significa que ahora el sistema logra capturar más casos de ciudadanos que efectivamente cambiaron sus hábitos.

**5.3 Conclusión**

La optimización de hiperparámetros permitió obtener modelos con **mayor recall y F1-score**, reforzando la utilidad de los ensambles (Random Forest y XGBoost) para abordar problemas de seguridad ciudadana.  
Si bien el valor de ROC-AUC aún se encuentra en torno a 0.68 (moderado), la priorización de Recall asegura que el modelo cumpla el objetivo de identificar de manera eficaz a la población más vulnerable frente a la delincuencia.

**6. Análisis de sesgo y varianza**

El análisis de sesgo y varianza permitió identificar el comportamiento de los modelos respecto a su capacidad de generalización:

* **Árbol de Decisión:** presentó un **alto riesgo de sobreajuste**, con varianza elevada al ajustarse demasiado a los datos de entrenamiento.
* **Regresión Logística:** mostró **subajuste** (sesgo alto), con desempeño limitado al no capturar adecuadamente la complejidad de las relaciones entre variables.
* **Random Forest:** alcanzó un desempeño equilibrado con valores similares en entrenamiento (0.663) y prueba (0.640), lo que refleja buena capacidad de generalización.
* **XGBoost:** también mostró equilibrio, con un rendimiento en entrenamiento (0.666) y prueba (0.642), lo que indica un ajuste robusto sin sobreajuste significativo.

**Conclusión**

La optimización de hiperparámetros permitió reducir el sobreajuste en los modelos de ensamble, mejorando su estabilidad y asegurando un balance adecuado entre sesgo y varianza. Tanto **Random Forest** como **XGBoost** se consolidan como las mejores opciones, al ofrecer un **compromiso óptimo entre ajuste y generalización**.

**7. Visualizaciones y análisis estadístico**

Las visualizaciones permitieron complementar el análisis de los modelos y del comportamiento de la variable objetivo:

* **Histogramas por edad y sexo:**
  + Se observó que los cambios de hábito (CAMBIO\_HABITO=1) son más frecuentes en población joven-adulta, con una tendencia decreciente hacia edades mayores.
  + En la distribución por sexo, tanto hombres como mujeres reportaron cambios, aunque con mayor proporción en mujeres.
* **Gráfico de importancia de variables (Random Forest optimizado):**
  + La **edad** resultó ser el predictor más influyente.
  + Le siguieron variables asociadas a victimización (P601), confianza en la policía y estrato socioeconómico.
  + Estas variables destacan el rol de la experiencia directa de inseguridad y la percepción institucional en la decisión de modificar hábitos.
* **Curvas ROC y Precision-Recall:**
  + Los modelos optimizados (Random Forest y XGBoost) alcanzaron un desempeño superior frente a los modelos básicos, con curvas más alejadas de la diagonal aleatoria.
* **Matrices de confusión:**
  + Se evidenció una reducción de falsos negativos en los modelos optimizados, lo cual es clave en este contexto, pues implica mayor capacidad de detectar ciudadanos que efectivamente cambiaron hábitos.

**8. Conclusiones y recomendaciones**

El estudio permitió desarrollar y evaluar modelos de Machine Learning para predecir el **cambio de hábitos por temor a la delincuencia en Lima Metropolitana**, empleando datos de la ENAPRES 2023 (Capítulo 600).

**Conclusiones**

* Los **modelos de ensamble (Random Forest y XGBoost)** fueron los más adecuados, alcanzando mejores métricas de Recall y F1-score en comparación con modelos básicos como Árbol de Decisión o Regresión Logística.
* Los **factores más influyentes** en la predicción fueron:
  + Percepción de inseguridad en el barrio.
  + Confianza en la policía.
  + Edad y sexo.
  + Experiencias de victimización.
* Se identificó que **mujeres y adultos mayores** presentan mayor propensión a modificar sus hábitos ante la percepción de inseguridad.

**Recomendaciones**

* **Campañas focalizadas:** orientar intervenciones en mujeres y adultos mayores, quienes muestran mayor vulnerabilidad y tendencia a cambiar conductas.
* **Integración tecnológica:** implementar el modelo en entornos de **Big Data** (Hadoop, Spark) para escalar el análisis y procesar información en tiempo real.
* **Visualización interactiva:** usar herramientas como **Power BI** o **Looker Studio** para que autoridades y tomadores de decisiones puedan explorar los resultados en dashboards dinámicos.

**9. Anexos**

* **Código Python completo:** se adjunta el archivo en formato .ipynb (Jupyter/Colab) y .py, para que pueda ejecutarse o revisarse directamente desde la PC sin necesidad de conexión a Colab.
* **Resultados exportados:** se incluyen los archivos CSV con métricas y predicciones generadas durante el proceso (validación cruzada y modelos optimizados).
* **Gráficas:** se adjuntan las visualizaciones principales (ROC, Precision-Recall, importancia de variables, histogramas y matrices de confusión).