Proyecto de Machine Learning para la predicción de la adopción de mascotas en refugios

Ana Isabel Patiño Osorio*, Cristian Daniel Muñoz Botero†
Departamento de Ingeniería de Sistemas, Universidad de Antioquia
Medellín, Colombia
Email: *anai.patino@udea.edu.co, †cristian.munoz3@udea.edu.co

***Resumen*—Este trabajo explora la aplicación de técnicas de Machine Learning para predecir la probabilidad de adopción de mascotas en refugios, utilizando un conjunto de datos que incluye características clave de cada animal. El objetivo de esta herramienta es asistir a las organizaciones protectoras de animales en la toma de decisiones informadas, enfocándose en promover la adopción de aquellos animales con menores probabilidades de ser adoptados.**

***Palabras Clave*——Machine Learning, adopción de mascotas, clasificación binaria**

I. Introducción

Cada año, miles de mascotas son acogidas por refugios, muchos de los cuales tienen recursos limitados para cuidar de ellas por tiempos prolongados. Identificar qué mascotas tienen más o menos probabilidades de ser adoptadas permite a estas instituciones enfocar mejor sus esfuerzos. Este trabajo propone el uso de un modelo de Machine Learning que analice los atributos de cada animal para estimar su probabilidad de adopción.

II. Descripción del problema

La adopción de mascotas desempeña un papel fundamental en nuestra sociedad, ya que brinda una nueva oportunidad de vida a animales que han sido abandonados o rescatados de situaciones que comprometían su bienestar. Sin embargo, muchos refugios enfrentan serios desafíos, como la sobre población, lo que dificulta una atención adecuada y eleva los costos operativos. Además, la permanencia prolongada de los animales en estos centros puede afectar negativamente su calidad de vida.

Debido a lo anterior, el desarrollo de una solución basada en técnicas de Machine Learning resulta altamente valioso. Esta permitiría predecir la cantidad de días que pasará una mascota en un refugio según sus características. Por ejemplo, verificar si una mascota vacunada es propensa a ser adoptada más rápido que una mascota que no cuente con sus vacunas al día. Esto facilitaría la implementación de estrategias más efectivas de promoción permitiendo priorizar campañas dirigidas a aquellos animales con mayor probabilidad de tener estancias muy largas en un albergue o incluso, no ser adoptados. De esta manera, se optimizan los recursos disponibles y se incrementan las posibilidades de que más animales encuentren un hogar.

La base de datos para predecir el estado de adopción de mascotas contiene información detallada sobre animales disponibles para adopción en refugios. A continuación presentamos una breve descripción:

Número de muestras: 2007

Número de variables: 13

- PetID: Identificador único de cada animal.
- PetType: Tipo de mascota (Perro, gato, etc).
- Breed: Raza específica de cada mascota.
- AgeMonths: Edad de la mascota en meses.
- Color: Color de la mascota.
- Size: Categoría del tamaño de la mascota (Pequeño, Mediano, Grande).
- WeightKg: Peso de la mascota en kilogramos.
- Vaccinated: Estado de vacunación de la mascota (0 – No vacunado, 1 – Vacunado).
- HealthCondition: Si presenta una condición de salud (0 - Saludable, 1 - Condición médica).
- TimeInShelterDays: Tiempo que la mascota ha estado en el refugio en días.
- AdoptionFee: Tarifa de adopción que se cobra por la mascota en dólares.
- PreviousOwner: Si la mascota tuvo un dueño anterior (0 - No, 1 - Sí).
- AdoptionLikelihood: Si la mascota fue adoptada o no (0 - No adoptada, 1 - Adoptada).

No hay existencia de datos faltantes.

Tipo de codificación usado para las diferentes variables del problema:

Variable	Tipo	Codificación
PetType	Categórico nominal	One hot encoding
Breed	Categórico nominal	Label encoding
AgeMonths	Numérico	Normalización
Color	Categórico nominal	One hot encoding
Size	Categórico ordinal	Ordinal encoding
WeightKg	Numérico	Normalización
Vaccinated	Numérico binario	—
HealthCondition	Numérico binario	—
TimeInShelterDays	Numérico	Normalización
AdoptionFee	Numérico	Normalización
PreviousOwner	Numérico binario	—
AdoptionLikelihood	Numérico binario	—

Cuadro I
Descripción de variables y su codificación

Se optó por utilizar el paradigma de aprendizaje supervi sado, dado que se cuenta con un conjunto de datos etiquetado. En este caso, se seleccionó la variable AdoptionLikelihood

como variable objetivo, con el propósito de predecir la probabilidad de adopción a partir de las características del animal. Esta información es valiosa para enfocar los esfuerzos en aquellos animales con menor probabilidad de ser adoptados, optimizando los recursos disponibles en los refugios.

III. Estado del arte

III-A. Sistema de recomendación para adopción de animales PetMatch[1]

Este trabajo de Nathalie Estrella presenta un sistema de recomendación híbrido que combina filtrado colaborativo y basado en contenido. Se utilizan técnicas como Random Forest para clasificación y una red neuronal convolucional (CNN) para analizar características visuales de las mascotas.

La validación del sistema incluyó métricas como exactitud, recall y similitud coseno, que mide la similitud entre vectores de preferencias del adoptante y características de la mascota:

$$Sim(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

Valores cercanos a 1 indican alta similitud.

Inicialmente se usó KNN, con una precisión del 33.31 % mejorada hasta 43.05 % (similitud promedio 0.78). Random Forest incrementó la precisión a 44.91 % y similitud a 0.99. Finalmente, el modelo XGBoost alcanzó una precisión del 94.46 % y un recall del 92.37 %, destacándose como la mejor solución.

III-B. Increasing adoption rates at animal shelters: a two-phase approach to predict length of stay and optimal shelter allocation[2]

En este trabajo, desarrollado por Janae Bradley y Suchithra Rajendran, se usó un paradigma de aprendizaje supervisado, tratando de predecir la duración de la estancia de los animales en el refugio según varias características: el tipo de animal, edad, género, raza, tamaño y ubicación del refugio. Se usaron como técnicas de aprendizaje la regresión logística, XGBoost, Random Forest y redes neuronales artificiales. Para el entrenamiento y validación de los mismos no se menciona dentro del texto una metodología de validación explícita. Simplemente se menciona que se dividen los datos en datos de entrenamiento y test y las métricas que se usaron para evaluar dichos modelos fueron: Recall, precisión y F_β score con $\beta = 1$.

Generalmente, se pudo ver que para predecir tiempos cortos de estancia, aunque se tuvo un rendimiento bajo en todos los modelos, el que más destacó fue el Random Forest; para estancias medias todos los modelos tuvieron un bajo rendimiento; para estancias largas el Gradient Boosting mostró un excelente desempeño lo mismo que para estancias muy largas (mascotas sacrificadas), en los que el recall y el F1 score alcanzaron los valores de 70.96 % y 74.77 % tanto para Gradient Boosting como para Random Forest.

En promedio, el algoritmo de Gradient Boosting obtuvo un desempeño cercano al 60 % en todas las métricas, mientras que la regresión logística fue el peor. Por último, se hizo un análisis de características en el cual se pudo observar que la edad avanzada, el tamaño de la mascota y el color fueron factores clave en la predicción de la duración de la estancia. Las demás variables no tuvieron un impacto significativo.

III-C. Predicting Pet Adoption Outcomes: A Comparative Study of Machine Learning Models[3]

Este estudio se realizó usando el mismo dataset con el que se quiere abordar el problema. En el artículo se expresa que se quiere usar un paradigma de aprendizaje supervisado, ya que se quiere predecir la etiqueta de salida, la cual ya tenemos. Las técnicas usadas por los investigadores del trabajo fueron: Artificial neural networks, Logistic regression, Decision trees y Random Forest. No se menciona una metodología de validación específica usada en este trabajo. La principal métrica usada en el trabajo fue el Accuracy, y la otra métrica usada fue el Area Under The Curve (AUC) de la curva ROC. Para las redes neuronales artificiales se pudo observar sobreajuste. Se obtuvo un accuracy del 99 % en train y 89 % en test y para la AUC se obtuvo 99 % en train y 95 % en test. Indican que se debe a que las redes neuronales requieren un dataset más amplio y una correcta regularización. Usando Random Forest se obtuvo menos diferencia en el accuracy, solo 91.29 %. Sin embargo se observa una oportunidad de mejora en el ajuste dado que la diferencia en el AUC es de 4 % aproximadamente. Los árboles de decisión fueron la mejor opción de todas con 92.53 % de accuracy en test y una diferencia de accuracy del 2.74 %. La diferencia del AUC es de solo 1.13 %. Por último, la regresión logística fue la que obtuvo peores métricas con un accuracy en test de 88.55 %.

III-D. Pet analytics: Predicting adoption speed of pets from their online profiles[4]

En este trabajo se optó por la metodología de trabajo supervisada, ya que se consideró como target la variable de adoption speed la cual se encuentra en el dataset que se usó. En este trabajo se usó: Regression, Least Angle Regression, Decision Tree, Gradient Boosting y Neural Networks, Random Forest (HP Forest) y modelo Ensemble donde combina todos los anteriores excepto HP Forest. Se dividieron los datos en 70 % para training y 30 % para testing, pero no se menciona una metodología de validación específica. La métrica que se usó en el estudio es la métrica ASE (Average squared error). La cual se define como:

$$ASE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y_i} - y_i)^2$$

Con algunos modelos no se usó la información textual excepto con el ensemble. El rendimiento de los modelos, medido con el ASE arrojó los siguientes resultados: XGBoost con información textual y Ensemble arrojaron un ASE de 1.15584, siendo los que menor ASE obtuvieron. Seguido están el XGBoost sin información textual, el árbol de decisión y las redes neuronales sin información textual.

IV. Entrenamiento y evaluación de modelos

Durante todo el trabajo, se utiliza una técnica de Stratified K-Folds Cross-validation con 5 folds. Esto dado que la variable de salida se encuentra desbalanceada, y queremos que cada fold guarde las proporciones entre las clases para hacer la validación. La clase 0 tiene 1348 representantes y la clase 1 cuenta con 659.

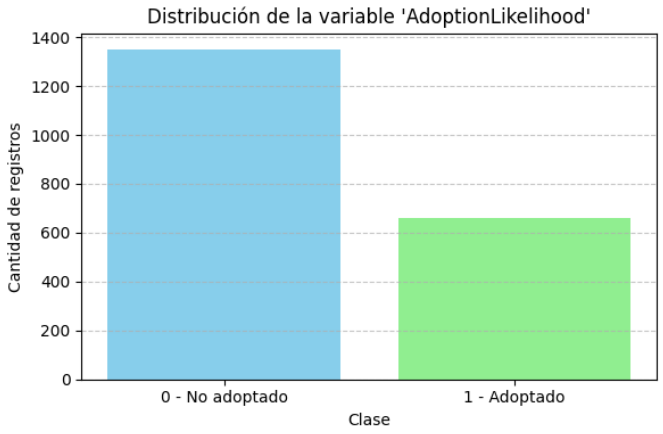

Figura 1. Desbalance de la clase objetivo

Se utilizaron 5 modelos de aprendizaje diferentes: Logistic regression, KNN, Random Forest, MLP y SVM.

IV-A. Análisis de hiperparámetros analizados para cada modelo

IV-A1. Modelo KNN: Para el modelo K-Nearest Neighbors, se evaluaron los hiperparámetros mostrados en la Tabla II.

Hiperparámetro	Valores evaluados
`n_neighbors`	1, 2, 3, ..., 30
`weights`	`'uniform'`, `'distance'`
`metric`	`'euclidean'`, `'manhattan'`

Cuadro II
Hiperparámetros evaluados para el modelo KNN

Esta malla de hiperparámetros generó 120 combinaciones posibles.

IV-A2. Modelo Random Forest: Para el modelo Random Forest, se analizaron los hiperparámetros mostrados en la Tabla III.

Hiperparámetro	Valores evaluados
`n_estimators`	50, 100, 150, 200
`max_depth`	`None`, 10, 15, 20
`min_samples_split`	2, 5, 10, 15, 20
`min_samples_leaf`	1, 2, 5, 10, 15, 20
`max_features`	`'sqrt'`, `'log2'`
`bootstrap`	`True`
`criterion`	`'gini'`, `'entropy'`

Cuadro III
Hiperparámetros evaluados para el modelo Random Forest

Esta malla de hiperparámetros generó 1920 combinaciones posibles.

IV-A3. Modelo MLP (Multi-Layer Perceptron): Para el modelo MLP, se generó una malla de hiperparámetros combinando los valores mostrados en la Tabla IV.

Hiperparámetro	Valores evaluados
`hidden_layer_sizes`	(32,), (64,), (64, 32), (64, 32, 16)
`activation`	`'relu'`, `'tanh'`
`alpha`	0.0001, 0.001
`learning_rate_init`	0.001, 0.01

Cuadro IV
Hiperparámetros evaluados para el modelo MLP

Esta malla de hiperparámetros generó 32 combinaciones posibles.

IV-A4. Modelo SVM (Support Vector Machine): Para el modelo SVM con clasificador SVC, se evaluaron los hiperparámetros mostrados en la Tabla V.

Hiperparámetro	Valores evaluados
`C`	0.1, 1, 10
`kernel`	`'linear'`, `'rbf'`, `'poly'`
`gamma`	0.001, 0.01, 0.1, 1, `'scale'`, `'auto'`
`degree`	2, 3
`class_weight`	`None`, `'balanced'`
`max_iter`	-1 (sin límite)

Cuadro V
Hiperparámetros evaluados para el modelo SVM

Esta malla de hiperparámetros generó 216 combinaciones posibles.

IV-A5. Método de selección de hiperparámetros: Utilizando la librería `scikit-learn`, se experimentó con cada una de estas combinaciones mediante el objeto `GridSearchCV`. La evaluación se realizó usando la métrica **ROC AUC**, lo cual permitió identificar la mejor configuración de hiperparámetros para cada modelo.

- **K-Nearest Neighbors (KNN):**
  Mejores hiperparámetros encontrados: `{'metric': 'manhattan', 'n_neighbors': 26, 'weights': 'distance'}`.
  Mejor puntuación ROC AUC: **0.744**.
- **Random Forest:**
  Mejores hiperparámetros encontrados:
  `{'bootstrap': True, 'criterion': 'gini', 'max_depth': 15, 'max_features': 'sqrt', 'min_samples_leaf': 5, 'min_samples_split': 15, 'n_estimators': 100}`.
  Mejor puntuación ROC AUC: **0.826**.
- **Multilayer Perceptron (MLP):**
  Mejores hiperparámetros encontrados:
  `{'hidden_layer_sizes': (64, 32), 'activation': 'tanh', 'alpha': 0.001, 'learning_rate_init': 0.01}`.
  Mejor puntuación ROC AUC: **0.808**.
- **Support Vector Machine (SVM):**
  Mejores hiperparámetros encontrados:
  `{'C': 10, 'class_weight':`

```
None, 'coef0': 0.0,
'decision_function_shape': 'ovr',
'degree': 2, 'gamma': 0.001, 'kernel':
'linear', 'max_iter': -1, 'shrinking':
True, 'tol': 0.001}.
```
Mejor puntuación ROC AUC: **0.810**.

IV-B. Métricas usadas para evaluar los modelos

Las métricas escogidas para evaluar los modelos fueron la precisión, la sensibilidad y el ROC-AUC. Dado que la variable de salida presenta un desbalance considerable, con más muestras a la clase "No adoptado", una precisión alta indica que el modelo está generalizando de manera correcta y que está captando las relaciones clave que hacen que una mascota sea adoptada. Una sensibilidad alta, garantiza que el modelo reconozca mascotas que sí serán adoptadas realmente. Es importante porque una baja sensibilidad puede hacer pasar desapercibidas a mascotas que sí pueden ser adoptadas. El ROC-AUC es importante en nuestro problema, ya que por el desbalance necesitamos una métrica que mida qué tanta capacidad tiene el modelo de clasificar correctamente. Que le de más probabilidad a los adoptados que a los no adoptados.

V. Resultados

V-A. Regresión logística

Para la regresión logística obtuvimos los siguientes coeficientes ordenados en magnitud:

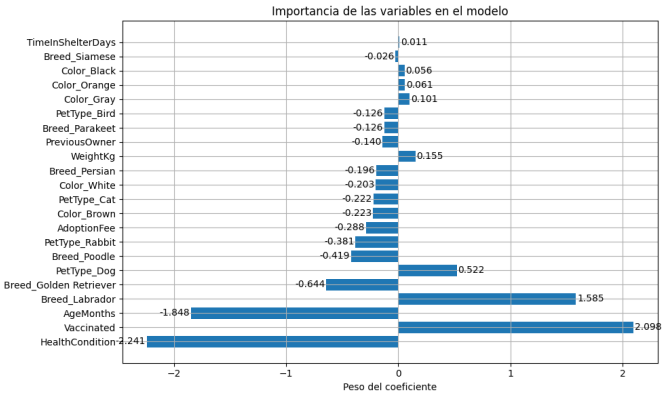

Figura 2. Pesos de los coeficientes en el modelo de Regresión logísitica después del entrenamiento

Las métricas obtenidas fueron:

Cuadro VI
Resultados del modelo de regresión logística en entrenamiento, validación y prueba

Conjunto	Precisión	Recall	ROC AUC
Entrenamiento	0.543 ± 0.005	0.759 ± 0.013	0.825 ± 0.003
Validación	0.533 ± 0.019	0.740 ± 0.027	0.810 ± 0.012
Prueba (Test)	0.505	0.727	0.787

Indicando una fuerte capacidad de clasificación y fuerte confianza al predecir a una mascota como adoptada.

V-B. KNN

Para el modelo KNN se obtuvieron los siguientes resultados:

Cuadro VII
Resultados del modelo KNN en entrenamiento, validación y prueba

Conjunto	Precisión	Recall	ROC AUC
Entrenamiento	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
Validación	0.635 ± 0.043	0.347 ± 0.060	0.752 ± 0.026
Prueba (Test)	0.539	0.311	0.706

Los resultados en entrenamiento dan cuenta de un fuerte sobreajuste del modelo.

El parámetro más importante para este modelo es el número de vecinos que se tiene en cuenta para el cálculo de la clase de la muestra. Se puede observar cual es el efecto de dicho parámetro sobre la métrica ROC-AUC cuando se estaba realizando la validación cruzada en la siguiente gráfica:

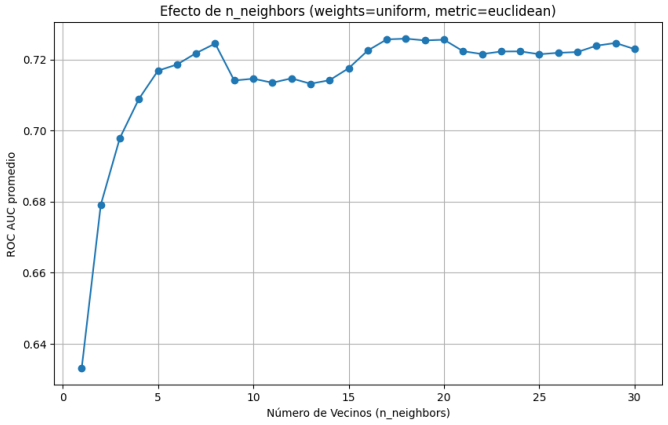

Figura 3. Efecto que tiene el hiperparámetro de número de vecinos sobre la métrica ROC AUC

Las métricas son similares a la de la regresión logística sin embargo, el recall presenta una fuerte caída. Indicando que el modelo falla más en detectar los casos de mascotas adoptadas verdaderos.

V-C. Random Forest

Para el modelo Random Forest se obtuvieron los siguientes resultados en métricas:

- Precisión: 0.850
- Recall: 0.386
- ROC AUC: 0.782

En general, el parámetro que más efecto tuvo sobre la métrica ROC AUC fue el número de estimadores, el cual se puede observar en esta gráfica:

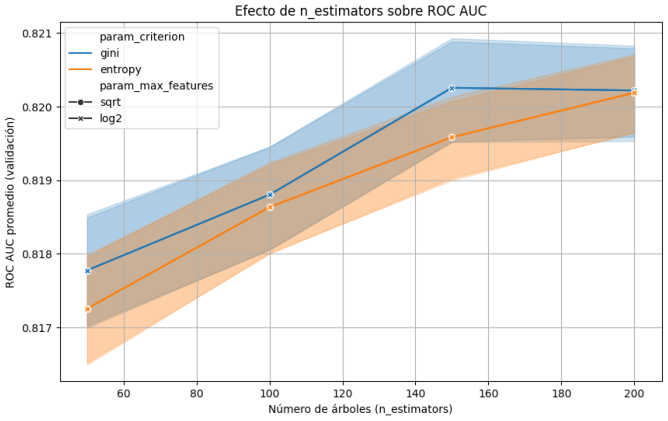

Figura 4. Efecto que tiene el hiperparámetro de número de árboles sobre la métrica ROC AUC

Se puede observar entonces que usando el parámetro de criterio 'gini' se converge más rápido a un máximo de ROC AUC, específicamente en 150 estimadores. Mientras que con el criterio 'entropy' alcanzar este máximo de ROC AUC toma 200 estimadores.

V-D. MLP

Con el modelo de MLP obtuvimos las siguientes métricas:

Cuadro VIII
RESULTADOS DEL MODELO MLP EN ENTRENAMIENTO, VALIDACIÓN Y PRUEBA

Conjunto	Precisión	Recall	ROC AUC
Entrenamiento	0.747 ± 0.045	0.428 ± 0.071	0.813 ± 0.018
Validación	0.709 ± 0.063	0.384 ± 0.071	0.785 ± 0.026
Prueba (Test)	0.875	0.371	0.793

En general, se pudo observar la influencia de los hiperparámetros de capas ocultas y learning rate sobre la métrica ROC AUC fue mayor que para los demás hiperparámetros, a los cuales dicha métrica no cambiaba significativamente. Se puede observar entonces las siguientes gráficas para comprender el efecto de los mismos sobre el ROC AUC:

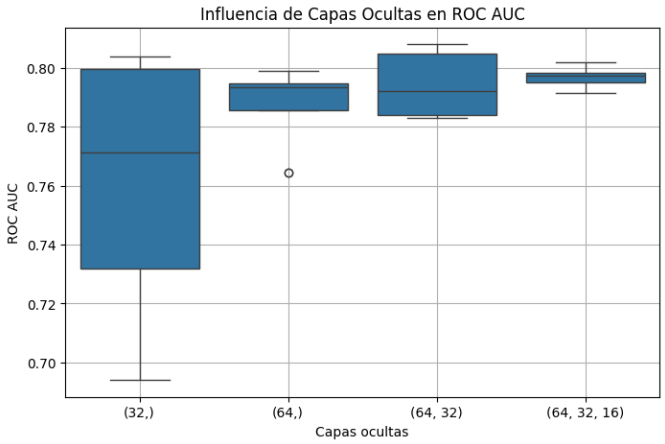

Figura 5. Efecto que tiene el hiperparámetro de capas ocultas sobre la métrica ROC AUC

Se puede entonces notar rápidamente que al agregar capas ocultas se agrega estabilidad a la métrica ROC AUC. Siendo particularmente notorio en el primer caso donde tenemos un rango intercuartílico bastante grande y el último caso, donde tenemos un rango intercuartílico muy pequeño, y una media más alta que para las otras configuraciones. Esto indica estabilidad y claramente preferimos un ROC AUC alto.

En la siguiente imagen se puede observar el efecto de Learning rate init sobre el AUC:

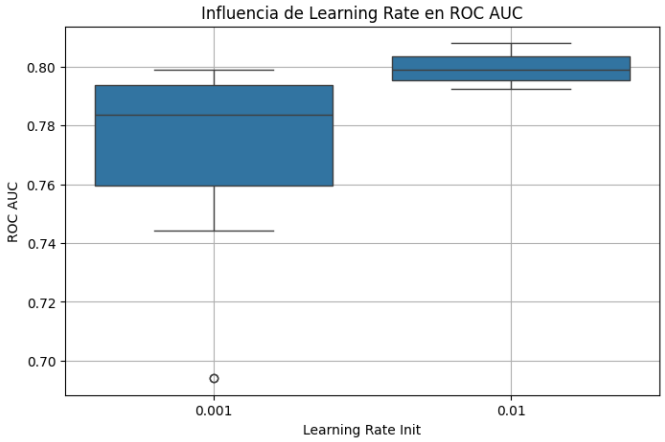

Figura 6. Efecto que tiene el hiperparámetro de learning rate init obre la métrica ROC AUC

Se puede observar en la gráfica un comportamiento parecido al de la gráfica 3. En este al aumentar el learning rate init a 0.01 se disminuye de manera abrupta el tamaño del rango intercuartílico y sube la media de los ROC AUC. Esto nos permite entonces, escoger dicho hiperparámetro dado que queremos un ROC AUC alto. El poco rango intercuartílico nos garantiza que estará muy cerca a ese valor con nuevas muestras.

V-E. SVM

Con el modelo SVM se obtuvieron las siguientes métricas:

Cuadro IX
DESEMPEÑO DEL MODELO SVM EN LOS CONJUNTOS DE ENTRENAMIENTO, VALIDACIÓN Y PRUEBA.

Conjunto	Precisión	Recall	ROC AUC
Entrenamiento	0.747 ± 0.045	0.428 ± 0.071	0.813 ± 0.018
Validación	0.510 ± 0.030	0.539 ± 0.041	0.780 ± 0.016
Prueba (Test)	0.660	0.530	0.781

El modelo presenta un desempeño general sólido en términos de su capacidad de discriminación (medida por el ROC AUC), tanto en validación como en prueba, con valores cercanos a 0.78, lo cual indica que el modelo es capaz de distinguir razonablemente bien entre las clases positivas y negativas.

En general, se pudo observar que los hiperparámetros que más influencia tenían sobre el ROC AUC eran el tipo de kernel usado, el gamma y el peso de las clases.

El efecto sobre el ROC AUC del tipo de kernel se puede observar en esta gráfica:

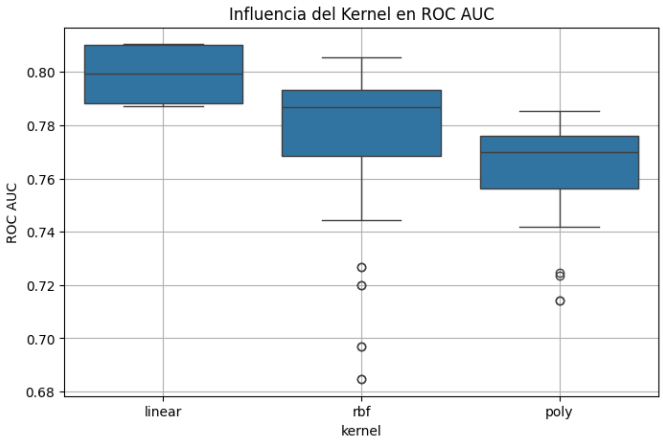

Figura 7. Efecto que tiene el hiperparámetro de kernel sobre la métrica ROC AUC

Se puede entonces observar que todos los tipos de kernel dan medias de ROC AUC diferentes, siendo la mejor el kernel lineal. Se puede observar que en general el kernel polinomial es el que menor índice de ROC AUC genera. Ninguno produce más estabilidad en la métrica que los demás. Todos tienen un rango intercuartílico similar en rango.

El gamma también tiene un efecto interesante en el ROC AUC:

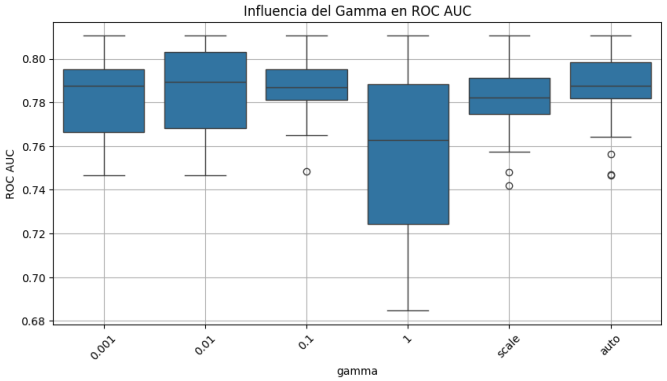

Figura 8. Efecto que tiene el hiperparámetro de gamma sobre la métrica ROC AUC

Se puede entonces observar que el gamma no tiene un comportamiento lineal sino que a medida que crece su comportamiento varía de diferentes formas. El gamma que genera una media de ROC AUC más alta es el 0.01 pero también se mueve en un rango más alto al ser más inestable. Un gamma de 0.1 parece que deja una métrica de ROC AUC más estable y con una media relativamente alta. Podemos observar también que para un gamma = 1 obtenemos la media de ROC AUC más baja y con mucha inestabilidad.

Por último, observemos el efecto del peso de las clases (class weight) sobre la métrica ROC AUC:

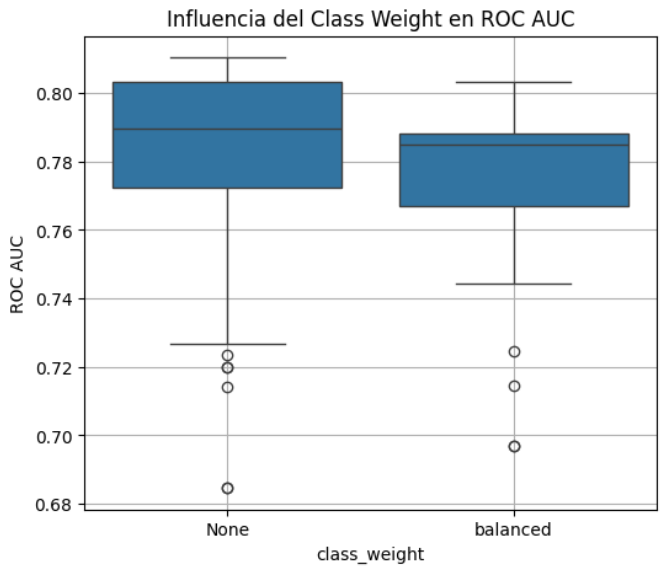

Figura 9. Efecto que tiene el hiperparámetro de class weight sobre la métrica ROC AUC

Se puede observar que al escoger un peso balanceado, la media de la métrica de ROC AUC baja en apróximadamente un 0.5 %. No es un cambio significativo pero se puede observar que cuando no se balancean los pesos se tiene un rango intercuartílico más estable al rededor de la media. En ambos casos se observan algunos outliers hacia abajo.

Para resumir los resultados de todos los modelos evaluados, se presenta entonces la siguiente tabla:

Cuadro X
RESUMEN DE MÉTRICAS PARA LOS MODELOS EVALUADOS

Modelo	Precisión	Recall	ROC AUC
Regresión Logística	0.505	0.727	0.787
KNN	0.539	0.311	0.706
Random Forest	0.850	0.386	0.782
MLP	0.875	0.371	0.793
SVM	0.660	0.530	0.781

VI. SELECCIÓN DE CARACTERÍSTICAS

En un primer momento, se realizó un análisis de correlación de las variables que se tenían en el dataset. El análisis arrojó la siguiente matriz de correlación:

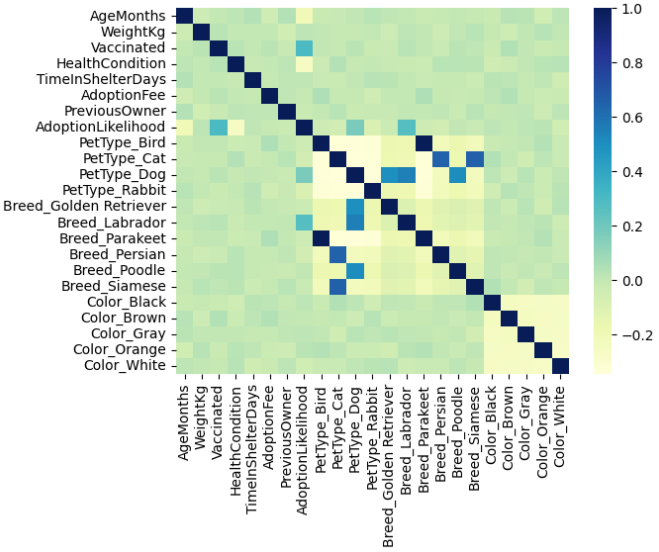

Figura 10. Matriz de correlación de los datos

Las variables que visiblemente tienen más correlación con la variable objetivo son: Vaccinated, PetType_Dog y Breed_Labrador

Luego, se realiza el algoritmo de selección de características en el conjunto de datos principal. Se lleva a cabo el algoritmo de Sequential Forward Selection con el criterio de la información mutua. Por lo tanto, empezamos con el Dataset vacío y agregamos columnas, teniendo en cuenta que siempre esta columna debe ser la que más información mutua tenga. Se para en este caso en 7 características. Los resultados de la información mutua los obtuvimos calculando la información mutua de un pequeño conjunto de características. Luego se agregaban nuevas características y las ordenamos de mayor a menor aporte de información. Definimos el umbral de información mutua acumulada en 0.225

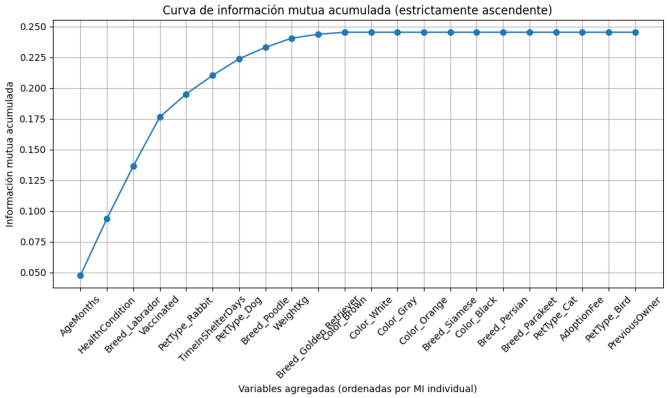

Figura 11. Gráfica de Información mutua acumulada

La función criterio (Información mutua) se escoge porque a diferencia de otras métricas como la distancia entre clases (diferencia de medias) o distancias probabilísticas (divergencias entre distribuciones) no requerimos supuestos fuertes sobre la forma de los datos y captura dependencias no lineales entre las variables y la clase objetivo.

Después de la aplicación del algoritmo al dataset completo nos quedamos entonces con las siguientes características: 'Vaccinated', 'AgeMonths', 'HealthCondition', 'TimeInShelterDays', 'Breed_Labrador', 'PetType_Rabbit', 'PetType_Dog' logrando una reducción del dataset en un 68.18 %.

VI-A. Entrenamiento de Regresión logística con nuevos datos

Se realiza nuevamente un entrenamiento similar al que se realizó con la regresión logística, solo que ahora con los nuevos datos entregados por el algoritmo de SFS usando información mutua como función criterio. Los resultados de las métricas calculadas fueron las siguientes:

- Precisión: 0.626
- Recall: 0.583
- ROC AUC: 0.799

Estos resultados muestran una mejora respecto al entrenamiento de un modelo de regresión logística con las 22 características que teníamos anteriormente. Muestra una mejora en la precisión del 12.1 % aproximadamente y el ROC AUC en un 1 % aproximadamente. Sin embargo disminuye el recall en un 14.4 % aproximadamente. Esto nos indica que, aunque el modelo se volvió más preciso al clasificar positivamente, es decir, comete menos falsos positivos, también se volvió más estricto, dejando de identificar una parte significativa de los casos positivos reales (mascotas que sí fueron adoptadas), lo que se refleja en la caída del recall.

VI-B. Entrenamiento de SVC con nuevos datos

Se realiza el entrenamiento del modelo SVM que se hizo anteriormente y se calculan las métricas. Los resultados de las mismas fueron los siguientes:

- Precisión: 0.673
- Recall: 0.576

- ROC AUC: 0.802

Obteniendo la siguiente curva ROC

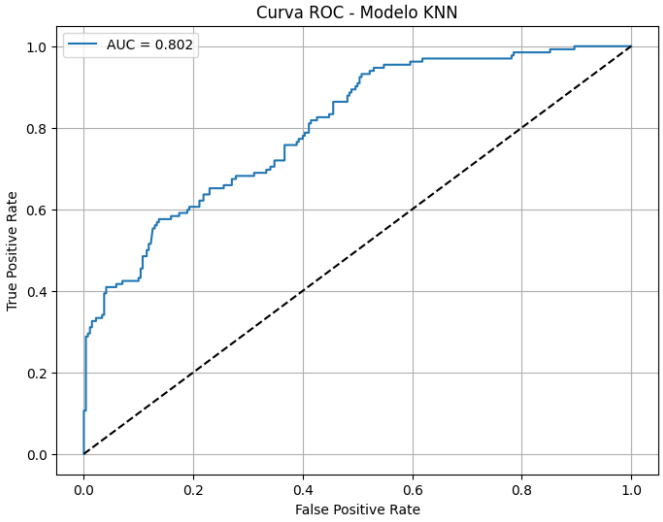

Figura 12. Curva ROC del SVC entrenado con los datos resultantes de aplicar SFS con información mutua

Como se puede observar el modelo mejoró un poco en Precisión, mejoró un poco también en recall y mejoró al rededor de un 2 % en ROC AUC. Por lo tanto, podemos concluir que el modelo realmente no necesita tantas características. La información discriminante realmente se concentra en unas pocas variables predictoras.

Cuadro XI
COMPARACIÓN DE DESEMPEÑO CON Y SIN SFS PARA LOS MODELOS DE REGRESIÓN LOGÍSTICA Y SVM

Modelo	**Precisión**	**Recall**	**ROC AUC**
Reg. Logística (original)	0.505	0.727	0.787
Reg. Logística (SFS)	0.626	0.583	0.799
SVM (original)	0.660	0.530	0.781
SVM (SFS)	0.673	0.576	0.802

VII. EXTRACCIÓN DE CARACTERÍSTICAS

Para realizar la extracción de características, primero se buscó el porcentaje de varianza explicada por cantidad de características. Podemos observar la siguiente gráfica:

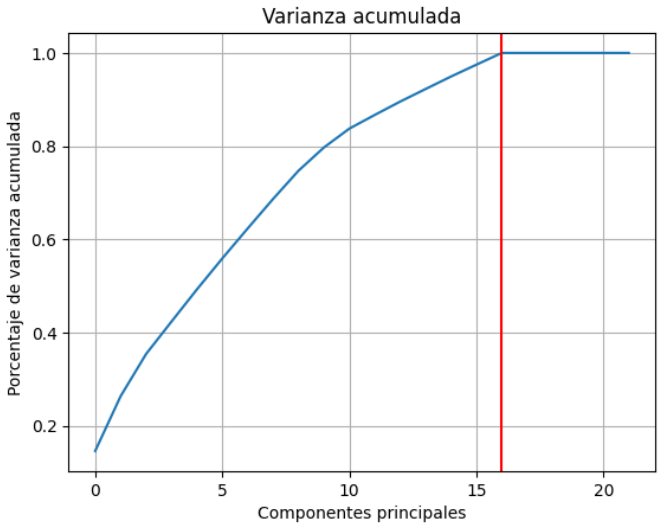

Figura 13. Porcentaje de varianza explicada acumulada por cantidad de características

En esta gráfica podemos observar que la varianza toma una forma muy peculiar y llega a su máximo en 16 características. Es aquí donde realizamos la eliminación de las demás 6 características logrando así una reducción del 27.27 % del dataset original.

VII-A. Entrenamiento de Regresión logística con los datos de PCA

Se realiza el entrenamiento del modelo de regresión logística con los datos arrojados por PCA y se calcula la precisión, el recall y el ROC AUC. Los resultados obtenidos son los siguientes:

- Precisión: 0.656
- Recall: 0.472
- ROC AUC: 0.803

Estos resultados dan cuenta de una mejora en la precisión en un 15 % aproximadamente, una disminución del 25.5 % del recall y un aumento del ROC AUC en un 1.6 %. Estos resultados dan cuenta de que el modelo se hizo más conservador. Solamente predice 1 cuando está más seguro, esto hace que deje pasar más casos que sí eran adopciones reales y por lo tanto se disminuya el recall. El pequeño aumento en el ROC da cuenta de que el modelo discrimina un poco mejor entre adoptados y no adoptados, pero no es un gran cambio.

VII-B. Entrenamiento de SVM con los datos de PCA

Se realiza el entrenamiento del modelo SVM con los parámetros óptimos encontrados en la sección 4 y se calcula la precisión, el recall y el ROC AUC. Los resultados obtenidos son los siguientes:

- Precisión: 0.720
- Recall: 0.407
- ROC AUC: 0.797

Estos resultados, comparados con los de la sección 5, dan cuenta de un aumento en la precisión de un 6 %, una

disminución en el recall del 12.3 % y un aumento en el ROC AUC del 1.6 %. Dichos cambios en las métricas son similares a los resultados con la regresión logística. El modelo se hizo más conservador y solo predice 1 cuando está más seguro, dejando pasar algunos casos que deberían ser adoptados.

Cuadro XII
COMPARACIÓN DE DESEMPEÑO CON Y SIN PCA PARA LOS MODELOS DE REGRESIÓN LOGÍSTICA Y SVM

Modelo	**Precisión**	**Recall**	**ROC AUC**
Reg. Logística (original)	0.505	0.727	0.787
Reg. Logística (PCA)	0.656	0.472	0.803
SVM (original)	0.660	0.530	0.781
SVM (PCA)	0.720	0.407	0.797

VIII. DISCUSIÓN Y CONCLUSIONES

VIII-A. Discusión

VIII-A1. Evaluación del desempeño de los modelos: La evaluación del desempeño de los modelos muestran un desempeño competitivo de los mismos aplicados a la predicción de adopción de mascotas. El análisis de los 5 modelos evaluados (Regresión logística, KNN, Random Forest, MLP y SVM) revela las diferencias entre los comportamientos de cada uno de estos:

El MLP obtuvo mayor precisión (0.875), seguido de Random Forest (0.850), lo que indica una alta confiabilidad cuando se predice que una mascota será adoptada. Sin embargo ambos modelos presentan un recall relativamente bajo (0.371 y 0.386 respectivamente), sugiriendo dificultades para identificar todas las mascotas que son realmente adoptadas. En contraste la regresión logística mostró el mejor recall (0.727) pero con menor precisión (0.505) evidenciando un comportamiento más sensible pero menos específico.

En términos de la métrica ROC AUC que es particularmente relevante dado el desbalance de clases del conjunto de datos, el MLP logró el mejor desempeño (0.793), seguido de cerca por la regresión logística (0.787) y RandomForest (0.782). Estos valores indican una buena capacidad discriminatoria, aunque aún se pueden mejorar.

VIII-A2. Impacto de la selección y extracción de características: Los experimentos con técnicas de reducción dimensional dieron a conocer detalles importantes sobre la naturaleza del problema. La aplicación de Sequential Forward Selection (SFS) con información mutua como criterio logró una reducción significativa del dataset (68.18 %) manteniendo solo 7 características de las 22 originales. Esta reducción mejoró la eficiencia computacional y también condujo a mejoras en el desempeño. La Regresión Logística con SFS mostró mejoras notables en precisión (0.626 vs 0.505) y ROC AUC (0.799 vs 0.787) y el SVM con SFS también presentó mejoras consistentes en todas las métricas evaluadas. Por otro lado, PCA con 16 componentes (reducción del 27.27 %) mostró resultados mixtos. Aunque mejoró la precisión en ambos modelos evaluados causó una disminución considerable en el recall, sugiriendo que la transformación lineal de PCA puede estar perdiendo información discriminante importante para la identificación de casos positivos.

VIII-A3. Comparación con el estado del arte: Al comparar nuestros resultados con los del trabajo de Zhang et al. podemos observar que el trabajo de ellos presenta la métrica de accuracy, por lo que no es directamente comparable con nuestras métricas. Sin embargo, dicho trabajo indica un accuracy de 92.53 % para árboles de decisión y 91.29 % para Random Forest. Nuestros resultados muestran que nuestro Random Forest alcanzó una precisión de 0.850 y ROC AUC de 0.782 lo que sugiere un desempeño robusto y conssistente con los hallazgos del trabajo de Zhang et al. Con el estudio de Bradley y Rajendran debido al enfoque diferente de su trabajo no podemos comparar directamente, pero ellos reportan un excelente desempeño con XGBoost. Nuestro trabajo, muestra resultados comparables con Random Forest alcanzando una precisión de 0.850 validando la efectividad de este tipo de algoritmos en este tipo de problemas.

Ahora, comparando con PetMatch el sistema híbrido reportado alcanzó 94.46 % de precisión más un 92.37 % de recall con XGBoost. Nuestros resultados, aunque no alcanzan ese nivel de métricas también demuestran la viabilidad de enfoques más simples e interpretables con nuestro mejor modelo (MLP) logrando 0.875 de precisión aunque con menor recall.

VIII-A4. Análisis de las variables más importantes: Los resultados de SFS identificaron las variables más discriminantes: 'Vaccinated', 'AgeMonths', 'HealthCondition', 'TimeInShelterDays', 'Breed _Labrador', 'AdoptionFee', y 'Color_Gray'. Estos hallazgos son consistentes con la literatura previa, donde variables como edad, condición de salud y tiempo en refugio han mostrado ser predictores importantes de adopción.

VIII-B. Conclusiones

Nuestros 5 modelos alcanzaron una capacidad predictiva aceptable con ROC AUC superiores a 0.7 indicando que logran discriminar de buena manera entre clases. Las poca cantidad de muestras y el desbalance entre las clases, se mostrararon como problemas al evaluar los modelos y ver un patrón consistente, en el que los modelos con mayor precisión tendían a tener menor recall. También para la selección de características, en este problema específico SFS demostró ser más efectivo que PCA mejorando el desempeño mientras reduce notablemente la dimensionalidad. Este estudio puede proporcionar herramientas valiosas para los refugios de animales. permitiendo una identificación temprana de mascotas con baja probabilidad de adopción, optimizando recursos para enfocar esfuerzos en casos específicos y mejorar las estrategias de adopción de los refugios de animales basadas en las características más influyentes. La combinación de modelos interpretativos como la regresión Logística con técnicas de selección de características demuestra ser particularmente prometedora para aplicaciones del mundo real, donde tanto el desempeño predictivo como la interpretabilidad son cruciales para la adopción por parte de los usuarios finales.

REFERENCIAS

[1] N. B. E. Mejía, “Sistema de recomendación para adopción de animales "petmatch",” 2025.

[2] S. R. J. Bradley, “Increasing adoption rates at animal shelters: a two-phase approach to predict length of stay and optimal shelter allocation,” *BMC Veterinary Research*, 2021.

[3] R. Zhang, “Predicting pet adoption outcomes: A comparative study of machine learning models,” *Highlights in Science, Engineering and Technology*, 2025.

[4] A. Zadeh, K. Combs, B. Burkey, J. Dop, K. Duffy, and N. Nosoudi, “Pet analytics: Predicting adoption speed of pets from their online profiles,” *Expert Systems with Applications*, vol. 204, p. 117596, 2022.