Informe - Clasificación de Péptidos Antimicrobianos

# Descripción del problema

Encontrar nuevas moléculas antimicrobianas que puedan ser usadas contra bacterias resistentes a los antibióticos actuales se ha tornado en un problema farmacéutico mundial. Los péptidos antimicrobianos han tomado importancia en el desarrollo de nuevos antibióticos, dado que tienen grandes propiedades que actúan como medio de defensa en contra de enfermedades producidas por microorganismos.

Existe una gran cantidad de péptidos presentes en diferentes partes como plantas, mamíferos, insectos, entre otros [1]. Como hay tantos agentes presentes en la naturaleza y medio ambiente, su identificación se ha convertido en unos de los más grandes retos de la inteligencia artificial con miras a construir soluciones que apoyen la investigación y desarrollo de nuevas moléculas antimicrobianas.

# Metodología implementada

Para efectos de lograr identificar los péptidos antimicrobianos se implementaron tres diferentes técnicas de machine learning, Random Forest, Árbol de decisión y Regresión logística, teniendo en cuenta:

1. Uso de datos de entrenamiento con información de péptidos antimicrobianos y no antimicrobianos, adicionalmente datos de validación independientes a los de entrenamiento.
2. División de los datos de entrenamiento en dos, un subconjunto de datos utilizado netamente para entrenar los modelos de machine learning y otro para probar que tan efectivos son dichos modelos clasificando los péptidos.
3. Uso de métodos de tunning tales como GridSearchCV y RandomizedSearchCV, para determinar los hiperparámetros que dan mejor precisión los modelos.
4. Selección de características, teniendo en cuenta que se dispone de 1760 características relacionadas a los péptidos. Encontrar cuales de ellas describen mejor a péptidos con propiedades microbianas. La selección de características se realiza hallando el valor de la información (IV) usando el peso de la evidencia (WOE).
5. Comparar resultado de ejecutar los modelos con las 1760 características y solo con aquellas que describen mejor las propiedades microbianas.
6. Balanceo de los péptidos antimicrobianos y no antimicrobianos, con el fin de que sean comparables la dos clases y así obtener resultados más confiables.
7. Normalización de la información con el fin de que cada una de las características sean homogéneas entre sí.
8. Evaluar cada métrica otorgada por las técnicas de machine learning para concluir sobre cada una de ellas y su poder de clasificación.

# Análisis y discusión de los resultados

## Prueba 1. Bosques aleatorios

Bosques aleatorios o random forest por su nombre en inglés, es un método utilizado para resolver problemas de regresión y clasificación.

El bosque aleatorio es una herramienta de ensamble que toma un subconjunto de observaciones y un subconjunto de variables para construir un árbol de decisión. Construye varios de estos árboles de decisión y los une para obtener una predicción más precisa y estable.

### Hiperparámetros ajustados

**random\_state:** este parámetro hace que una solución sea fácil de replicar. Un valor definido de random\_state siempre producirá los mismos resultados si se da con los mismos parámetros y datos de entrenamiento, para el clasificador usamos 173.

**max\_features:** es la cantidad máxima de características que Random Forest puede probar en un árbol individual, el parámetro utilizado fue 0.025, lo que indica que el algoritmo utilizara el 2.5% del total de las variables para probar en cada árbol.

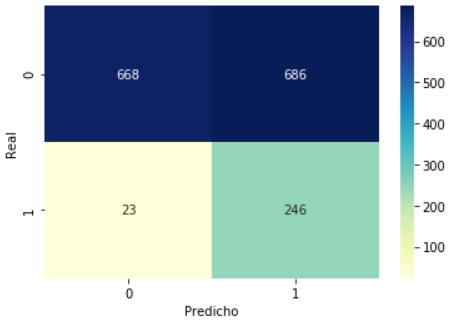
**n\_estimators:** es la cantidad de árboles que se desean construir antes de tomar la votación máxima o promedios de predicciones. Un mayor número de árboles le brinda un mejor rendimiento, pero hace que su código sea más lento, para el clasificador se usaron 38 árboles

**max\_depth:** es la altura de cada árbol, una buena altura reducirá la complejidad de los modelos aprendidos, reduciendo el riesgo de ajuste, el valor utilizado para el clasificador es de 1.9.

**criterion:** El valor usado para el clasificador es entropy. La entropía es una medida de impureza basada en la ganancia de información, la cual busca maximizar la información mutua, construyendo un nodo de probabilidad igual en cada árbol de decisión.

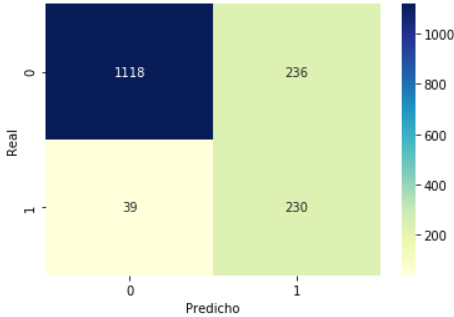
### Análisis de métricas

**Modelo sin ajustar Hiperparámetros y sin selección de variables**

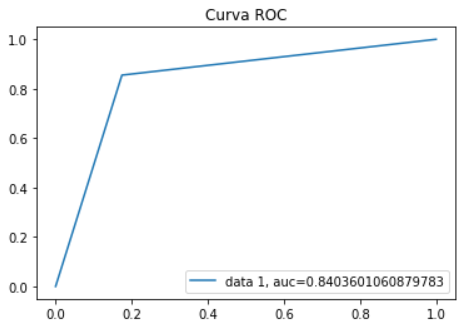


En etapa de entrenamiento y pruebas, se obtuvo muy buenos resultados. Sin embargo, al probar con los datos de validación, se encuentra que el rendimiento para predecir los péptidos antimicrobianos no es bueno, encontrando una precisión del 26%, esto se debe en gran media a que las clases en el conjunto de datos están desbalanceadas

**Modelo con ajuste de Hiperparámetros y con selección de variables**



Despues de entrenar el modelo con las variables seleccionadas (mas de 4 votos) y de probar con los datos de prueba y de validación, las métricas del modelo son mucho mejores. Al comparar la predicción de los péptidos antimicrobianos con los resultados que se obtuvieron sin ajustar los hiperparametros y sin seleccionar variables, observamos una precisión del 49.



Al verificar al área bajo la curva, observamos que el clasificador obtiene un 84%, lo que ubica el clasificador en el intervalo de valores para la curva roc como un clasificador "Bueno". Así mismo, encontramos un valor kappa de 52.6%, de lo cual se interpreta que el clasificador es mejor que un clasificador aleatorio.

### Tiempos de entrenamiento

## Prueba 2. Regresión logística

La técnica de regresión logística es un método de clasificación binaria

### Hiperparámetros ajustados

De acuerdo a la selección óptimo de hiperparámetros, *mejores parámetros son* solver: liblinear, penalt': l1, C: 0.001, dual: False, max\_iter: 100-110, para dar una precisión del 100%.

Sin embargo, realizando el ejercicio manual de probar con diferentes valores de los parámetros tenemos que los que mejoran el desempeño del modelo son los valores: **C = 0.0001, solver = 'liblinear', fit\_intercept= False**, donde:

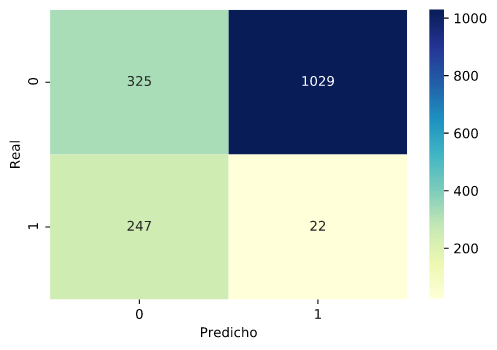
**Parámetro de regularización** (C = 1 / λ). Para valores pequeños de C, se aumenta la intensidad de la regularización, lo que creará modelos simples que no satisfacen los datos (no se ajustara). Para valores grandes de C, se reduce el poder de regularización, lo que implica que el modelo puede aumentar su complejidad y, por lo tanto, sobreajustar los datos. En este caso se toma un C = 0.0001 el cual es un número muy pequeño para aumentar la regularización y evitar el sobreajuste.

**Solver.** Algoritmo usado en el problema de optimización. Trata de encontrar el peso del parámetro que minimiza la función de costo. Liblinear es recomendado cuando se tiene alta dimensionalidad, lo cual aplica para este caso de estudio.

**Fit\_intercept**. Determina si se quiere incluir una constante β₀ al modelo. En este caso no se incluye.

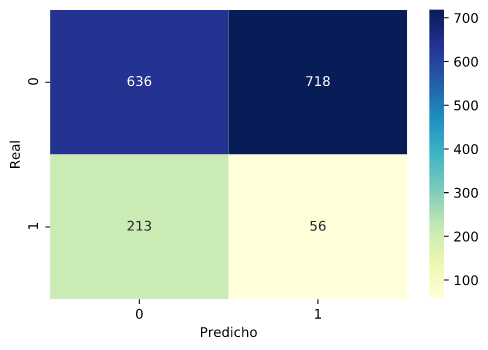
### Análisis de métricas

**Modelo sin ajustar los hiperparámetros**

Si bien la regresión logística en etapa de entrenamiento y prueba da muy buenos resultados, de un total de 2081 péptidos antimicrobianos que se tenían, el modelo predijo correctamente 2024, representando un 97% de la clase, de igual forma con los péptidos no antimicrobianos donde se tenían en total 1387 péptidos No antimicrobianos y el modelo indicó que efectivamente 1344 no lo eran. Cuando se analiza el modelo con los datos de validación, por su parte, el modelo no tiene un buen desempeño, esto se puede ver en las diferentes métricas generadas, donde todos los porcentajes son inferiores al 30%, esto puede darse porque las clases en este conjunto de datos están considerablemente desbalanceadas, por el alto número de características que existen y porque los parámetros de la regresión logística que vienen por defecto no son apropiados para las condiciones del dataset.

Adicionalmente, cuando se ve el coeficiente kappa, el cual indica que tan confiable es el modelo, de acuerdo a la tabla de validación del coeficiente, la regresión logística sin ajuste a los hiperparámetro, no es confiable.

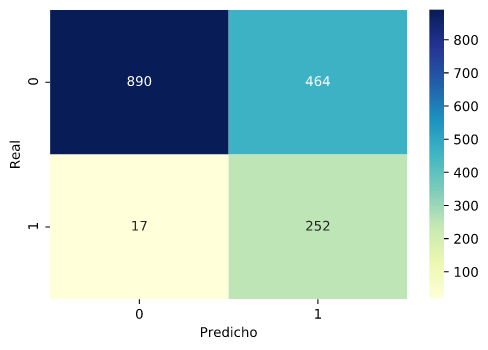
**Modelo sin selección de características ajustando hiperparámetros**



A pesar que en esta ejecución sobre los datos de validación mejora con respecto a la ejecución usando los hiperparámetros por defecto, de igual forma se tiene un performance bajo con los datos de validación, el accuracy en este caso es del 43%, lo que quiere decir que del total de datos solo el 43% fue capaz de predecirlo correctamente como péptidos antimicrobiano y péptidos no antimicrobiano. Con respecto a la predicción de los péptidos antimicrobiano sólo los predice en un 21%, caso contrario pasa con los datos de prueba, en donde los péptidos antimicrobiano son predichos en un 81% los no microbianos en un 96%.

En este modelo se vuelve a evidenciar un bajo coeficiente de kappa, concluyendo que el modelo no es confiable

**Modelo con selección de características ajustando hiperparámetros**



Una vez se ejecuta el modelo luego de haber seleccionado las características que tuvieran más de 6 votos, lo cual representa que son características con impacto en la clasificación de los péptidos antimicrobiano, se tiene que la regresión logística tanto en los datos de entrenamiento, como en los datos de pruebas y de validación tiene un comportamiento mucho mejor que a las pruebas previamente realizadas. Acá tenemos que de 269 péptidos antimicrobianos que se tienen, el modelo fue capaz de clasificar correctamente 252, lo que equivale al 94% de los datos de está clase. Por su parte cuando el modelo predice los péptidos antimicrobianos lo hace bien en un 35% de las veces, dando un precisión no muy buena.

De acuerdo al resultado del coeficiente de Kappa y la tabla de interpretación del coeficiente, se concluye que la regresión logística para clasificar los datos de validación para péptidos antimicrobianos es justa, es decir, no es la mejor pero contribuye a clasificar la clase.

### Tiempos de entrenamiento

## Prueba 3. Árbol de decisión

Bosques aleatorios o random forest por su nombre en inglés fue usado como clasificador

### Hiperparámetros ajustados

### Análisis de métricas

### Tiempos de entrenamiento

# Conclusiones

La reducción de la dimensionalidad cobra relevancia en los ejercicios de clasificación porque disminuye complejidad y permite que los modelos tengan un mejor comportamiento.

# Referencias

[1]. Péptido antimicrobiano. <https://es.wikipedia.org/wiki/P%C3%A9ptido_antimicrobiano#Clasificaci%C3%B3n>