# Parcial 2

Juan José Aguado, Emmanuel Collazos y Santiago Peña

[1] Pontifica Universidad Javeriana Cali
[2] Sistemas Inteligentes

**Abstract.** Se realizó una clasificación de datos y un análisis de distintos modelos de aprendizaje automático para determinar el reino de un organismo dado el numero de codones y la frecuencia de ciertos codones en su genoma. Se realizo un pre procesamiento y reducción de dimensionalidad de esto y a partir de esto se evaluó la predicción de 6 modelos en 2 corpus y sus variantes optimizadas por hiper parámetros. Se evaluaron por métricas como F1 score, precisión y Recall. Los resultados indican un mejor rendimiento del modelo K-Vecinos tanto en el corpus original como el que tuvo reducción de dimensionalidad.

**Keywords:** Clasificación · Modelo · Metricas.

## 1 Proceso

### 1.1 Descripción del corpus

El corpus es de uso de codones en el código genético de distintos organismos. Los codones son una secuencia de ADN o ARN de tres nucleótidos (3 letras como UUA o GCA) que forman una unidad de información genómica[2]. El corpus se conforma por 69 columnas y 13028 filas. El corpus se eligió de una base de datos de un repositorio[1].
Está conformado por 69 características para cada organismo; el reino al que pertenece (ej: planta, mamífero) dado en 3 caracteres de abreviación, el tipo de ADN del organismo que es un entero del 0 al 12 donde cada uno es una representación de la composición genómica del organismo (ej: 1-mitocondrial), el ID de la especie que identifica cada organismo según si ID original de la base de datos CUTG (ej: 100217), el numero de codones que es un entero que identifica la suma de las entradas de diferentes codones listados en el corpus (ej: 1995), el nombre de la especie siendo una cadena con el nombre científico de la especia (ej: Epizootic haematopoietic necrosis virus) y luego las columnas del 6 al 69 son codones y en cada fila hay la frecuencia de este en el organismo representado por un numero punto flotante con decimales de 5 digitos (ej: 0.01654,).
Hubo ciertos datos no válidos tales como el carácter "-" o datos enteros que se leían como cadenas los cuales fueron deshechos o reemplazados en el pre procesamiento, no habían datos nulos. El objetivo es tratar de determinar el reino con el resto de datos, los cuales varían dependiendo de la especia y familia a los cuales el organismo pertenezca, se trata de predecir utilizando más que todo la frecuencia de los codones (las columnas 6-9), además no tuvimos en cuenta datos innecesarios como el nombre del organismo o su ID pues no le dan información útil al modelo y solo serían datos que puedan darle un sesgo erróneo.

### 1.2 Pre procesamiento de datos

En un primer momento se separaron los datos en dos grupos, el primero solo tiene la columna de los reinos, y el otro los demás datos. A los reinos les aplicamos una transformación para que no

fueran cadenas de caracteres sino números enteros. Para el otro grupo eliminamos las columnas del nombre del organismo y del ID. También tuvimos que recorrer toda una columna que tenía los datos ínvalidos mencionados anteriormente, para cambiarlos por unos válidos o transformarlos a flotantes (antes eran cadenas de caracteres). Finalmente se separaron los datos en el grupo de entrenamiento y en el grupo de prueba y se aplicaron las transformaciones para hacer el scaling de los datos.

### 1.3 Reducción de dimensionalidad

Antes de hacer la reducción se ejecutaron unos métodos conocidos como El "método del codo" y el "coeficiente de silueta" las cuales son dos técnicas para determinar el número óptimo de clústeres (k) en algoritmos de agrupación, como el k-medias. En el método del codo se ejecuta el algoritmo de agrupación para diferentes valores de k y se calcula la inercia. El punto donde la inercia comienza a disminuir más lentamente se considera el número óptimo de clústeres. En el coeficiente de silueta se calcula la similitud de cada punto con su clúster y el clúster más cercano diferente. El coeficiente de silueta promedio se usa para encontrar el valor de k que maximiza la calidad de los clústeres. Los métodos nos recomendaron 2, 4 o 10.
La reducción de los datos se hizo a través del algoritmo de k-medias el cual es utilizado para agrupar un conjunto de datos en clústeres basados en similitudes entre los puntos de datos. Este proceso implica la inicialización de centroides, la asignación de puntos a clústeres en función de la cercanía a los centroides, el recálculo de los centroides como promedios de los puntos en cada clúster, y la repetición de estos pasos hasta que los centroides converjan a ubicaciones estables. La elección del número de clústeres (k) es crucial y puede influir en la estructura final de los clústeres. En este caso usamos un k=50, es decir, una amplia cantidad de clúster comparado a los resultados de los métodos aplicados. No obstante, se toma esta decisión para denotar una varianza notable en las métricas y los resultados de los diferentes modelos.

### 1.4 Modelos bases

**Regresión logística** La regresión logística es una técnica de clasificación que utiliza las matemáticas para encontrar relaciones entre 2 factores. En este caso se utilizó para predecir el reino dado los codones y el número de codones. Esta técnica dio una precisión del 86.83% en el corpus original y 85.65% en el corpus con clustering y también fue útil para sacar el R cuadrado, el cual indica la fuerza de la relación lineal entre variables [3] que va del 0 al 1, en este caso dio 0.74 lo que indica que es un modelo relativamente fiable para poder predecir la característica del reino.

**Naive Bayes** En este segundo modelo, se implementó el algoritmo de Naive Bayes, el cual "un clasificador probabilístico simple con fuerte suposición de independencia"[4]. Este dio una precisión de 54.48% en el corpus original el cual es mucho menor que el modelo anterior y muy bajo para considerarlo un modelo efectivo para esta clasificación. En contraste dio 94.97% de precisión en el corpus hecho con clustering por lo que a pesar de que no fue bueno en el normal es bastante preciso en otros casos por lo que no es un modelo que se deba dejar de lado.

**K-Vecinos** Los K-Vecinos, son "un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de

datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro."[5]. Este dio una precisión de 92.4% el cual es mejor que los otros modelos. Sin embargo, el valor de k y la métrica de distancia son factores clave en el rendimiento del algoritmo.

El k-Vecinos es simple pero efectivo y se utiliza comúnmente en problemas de clasificación y regresión, especialmente cuando se trabaja con conjuntos de datos relativamente pequeños o cuando se necesita una interpretación intuitiva del modelo. En el caso del nuevo corpus su precisión subió a 99.88% el cual mejora y sigue siendo el mayor comparado a los otros modelos con los nuevos corpus, lo cual muestra que sigue siendo el modelo más idóneo para estos este caso.

**Árboles de decisión** Los árboles de decisión son una técnica de aprendizaje automático utilizada para tomar decisiones basadas en una serie de condiciones o características que se evalúan en un conjunto de datos. Este dio una precisión del 78.58%, por lo que consideramos que tuvo un buen desempeño y que en general es un buen modelo para predecir los datos. Con el otro corpus tuvo un desempeño mucho mejor, logrando una precisión de 99.76%.

**Random Forest** El Random Forest es un modelo de aprendizaje automático que combina múltiples árboles de decisión para mejorar la precisión y evitar el sobreajuste. Se destaca por su capacidad de manejar datos de alta dimensionalidad, su resistencia al sobreajuste y su utilidad en una amplia variedad de aplicaciones, como clasificación de imágenes, diagnóstico médico y predicción de ventas. Este modelo dio una precisión de 90.21% el cual no destaca comparado a los otros modelos, es decir, que en este caso no era lo más recomendado. Sin embargo, es efectivo en la gestión de valores faltantes y datos desequilibrados. En el caso del nuevo corpus su precisión subió casi un 10% a 99.86% siendo la segunda mejor precisión de todos los modelos. Además, es de los modelos que tiene muy poca media de error cuadrático, lo que representa que las predicciones fueron fieles a los valores reales.

**Redes neuronales** Las redes neuronales están compuestas por unidades llamadas "neuronas" interconectados en capas, las cuales a traves de diferentes algoritmos de optimización tratan de realizar una tarea. Este dio una precisión del 84.26%, lo que indica que es un buen modelo que en la mayoría de los casos clasifica adecuadamente los datos. Con el otro corpus le fue mejor, alcanzando una precisión del 99.27%.

### 1.5 Modelos optimizados

En todos los modelos se utilizo grid search para la optimizacion y best params para encontrar los mejores hperparámetros.

**Regresión logística** Se utilizaron los hiperparámetros C(1 para el original y 10 para el corpus con clustering) y penalty(l2) para ambos corpus donde la precisión para el original fue de 86.72% (que fue peor que el modelo base por menos de un 1%) mientras para el corpus hecho con clustering fue de 96.21% que fue mucho mejor que el modelo base, la precisión es mas de un 10% mas alta.

**Naive Bayes** Se utilizaron los hiperparámetros priors(el mejor fue None) en el rango [None, [0.2, 0.8], [0.5, 0.5]] y varsmoothing(el mejor fue 1e-7) en el rango [1e-9, 1e-7] para ambos corpus donde la precisión para el original fue de 55.53% (que es mejor que el modelo base pero no significativamente) mientras para el corpus hecho con clustering fue de 94.91% que fue peor que el modelo base sin embargo no hay una diferencia ni siquiera del 0.1%.

**K-Vecinos** En el primer corpus se optimizaron los parámetros de número de vecinos n=3 en el rango [3, 5, 7, 9] siendo el menor, como método de ponderación w = weights, es decir por pesos, entre ese y distancia y como parámetro de distancia se escoge la Manhattan, entre esa y la Euclidiana. Al final la precisión mejora 0.41%, es decir, resulta en una precisión del 92.81%. No mejora mucho buscando una optimización. En el caso del otro corpus con el modelo optimizado, se escogen los mismos parámetros excepto el número de vecinos que cambia a 7. Pero, empeora su precisión, bajando a 99.76%, el cual no desmejora mucho, pero resulta mejor sin necesidad de buscar esos parámetros.

**Árboles de decisión** Para este modelo se optimizaron los hiperparámetros de criterion entre gini y entropy, siendo elmejor entropy; max depth en un rango entre 2 y 6, siendo el mejor 6; y splitter entre random y best, siendo el mejor best. La precisión fue de 69.07%, la cual es más baja que el modelo original, por lo que pensamos que pudo tener un sobreajuste que baje la precisión del modelo. Para el otro corpus se usaron los mismos rangos y la única diferecnia fue que en el splitter escogió random y la precisión subió a un 96.75%.

**Random Forest** En el primer corpus se optimizaron los parámetros de número de árboles n=500 en el rango [100, 200, 300, 500] siendo el mayor, como profundidad máxima de los árboles max = None, siendo el menor entre [None, 10, 20], como mínimo de muestras para dividir minSplit = 2 siendo la menor entre [2, 5, 7] y como mínimo de muestras en hojas minLeaf = 1, siendo el menor entre [1, 2, 4]. Al final la precisión empeora un 1.21%, es decir, resulta en una precisión del 89.10%. No empeora mucho. En el caso del otro corpus con el modelo optimizado, se escogen los mismos parámetros excepto el número de árboles que cambia al menor siendo 100. Pero, empeora su precisión, bajando a 99.67%, el cual no desmejora mucho, solo un 0.21%.

**Redes neuronales** Para este modelo se optimizaron los hiperparámetros de learning rate en un rango entre 0.0003 y 0.3, siendo el mejor 0.006452412194853618; n hidden en un rango entre 0 y 3, siendo el mejor 1, y n neurons en un rango entre 1 y 10, siendo el mejor 6. La precisión fue de 84.42%, la cual es ligeramente mejor que el modelo original, lo que indica que no se optimizó mucho el modelo. Con el otro corpus se usaron los mismos rangos, pero dieron valores distintos, los cuales son learning rate de 0.00783235216695384, n hidden de 0 y n neurons de 9. También aumento la precicisón hasta un 99.86%.

## 2 Resultados por modelo

| Modelo | Accuracy | Precisión | Recall | F1-Score | R-Cuadrado | MEC |
|---|---|---|---|---|---|---|
| Regresión Logística | 86.84% | 72.71% | 68.30% | 70.06% | 0.7435 | 3.249 |
| Naive Bayes | 54.48% | 48.14% | 56.46% | 42.02% | N/A | N/A |
| K-Vecinos | 92.40% | 78.02% | 78.26% | 77.97% | 0.8738 | 1.60 |
| Arboles de decisión | 78.58% | 62.16% | 59.29% | 60.3% | 0.5622 | 5.54 |
| Random Forest | 90.21% | 83.55% | 70.43% | 75.31% | 0.7912 | 2.65 |
| Redes Neuronales | 84.26% | 53.81% | 48.88% | 47.8% | 0.687 | 3.96 |

Table 1: Comparación de Modelos con Métricas Corpus Original

| Modelo | Accuracy | Precisión | Recall | F1-Score | R-Cuadrado | MEC |
|---|---|---|---|---|---|---|
| Regresión Logística | 85.65% | 47.28% | 55.01% | 48.73% | 0.6407 | 65.10 |
| Naive Bayes | 94.97% | 90.03% | 92.87% | 90.86% | N/A | N/A |
| K-Vecinos | 99.88% | 93.97% | 93.75% | 93.79% | 0.9974 | 0.48 |
| Arboles de decisión | 99.76% | 93.36% | 92.30% | 92.64% | 0.9946 | 0.9604 |
| Random Forest | 99.85% | 93.93% | 93.63% | 93.71% | 0.9972 | 0.50 |
| Redes Neuronales | 96.27% | 74.94% | 76.34% | 74.1% | 0.9194 | 14.58 |

Table 2: Comparación de Modelos con Métricas Corpus Reducido

## 3 Conclusiones y discusión

### 3.1 Comparación y conclusiones generales

Al analizar las diferentes métricas se puede concluir que en general a todos los modelos les fue bien a la hora de hacer las predicciones, siendo el más acertado el de K-Vecinos, seguido del Random Forest, y siendo el menos acertado el de Naive Bayes, que en resumen solo acertó la mitad de las veces, mientras que los demás dieron porcentajes más altos y errores más bajos. Para nosotros el más llamativo fue el de K-Vecinos, no solo por ser el más acertado y el que tuvo menos error, sino también porque sus métricas de clasificación fueron las que más cercanas estaban entre ellas (alrededor del 78%), menos la de accuracy, que dio 92%.

De estos resultados también se puede concluir que la predicción del reino de un organismo en función de las características que tomamos no es algo sin sentido, debido a que se puede evidenciar como los modelos logran ajustar de diferentes maneras los métodos para hacer las clasificaciones generalmente bien. Se puede decir que, si bien los diferentes organismos tienen varias cosas en común, hay muchas características que permiten separarlos por reino.

Finalmente podemos decir que los modelos que mejor se desempeñaron pueden ser bastante útiles para científicos que requieran de este tipo de clasificaciones para sus diferentes investigaciones, por lo que, si bien no son perfectos, son bastante acertados y solo requerirían de una pequeña supervisión para esos casos donde no clasifican los organismos adecuadamente.

### 3.2 Corpus procesados con clustering

La reducción de un corpus de texto puede influir en la precisión. Al reducir un corpus, se pueden obtener ventajas como un enfoque en datos relevantes, menor complejidad y un uso más eficiente de recursos. Sin embargo, también pueden surgir desventajas, como la pérdida de información, sesgos y limitaciones en tareas de aprendizaje profundo. La decisión de reducir un corpus debe basarse en los objetivos y las necesidades del proyecto, evaluando cuidadosamente el equilibrio entre precisión y la representatividad del corpus reducido.

En este caso podemos notar que la precisión en todos los modelos mejoro bastante, todos con un promedio de más de 96% mientras que antes de la reducción era de un promedio de 81% de precisión con los modelos. Las redes neuronales y la regresión logística tuvieron algunos datos que subieron la media del error cuadrático, por los problemas que supone la reducción de los datos. Sin embargo, los dos mejoraron su Accuracy. Por lo cual, en conclusión, al tener unos datos reducidos los modelos pueden predecir con más facilidad los valores reales, es decir, que los modelos se vuelven más confiables entre más sea la reducción del corpus.

### 3.3 Observaciones adicionales

Notamos que en todos los modelos la optimización era lo que más se demoraba y en varios casos no hubo mejoras significativas e incluso en algunos casos empeoraba por lo que recomendamos analizar que tanta precisión se requiere para la aplicación de algún modelo y así sopesar si la opción de optimizar vale la pena.

## References

1. Hallee,L. and Khomtchouk,B.B.. (2020). Codon usage. UCI Machine Learning Repository. https://doi.org/10.24432/C5KP6B.
2. “Codón”. Genome.gov. Accedido el 19 de octubre de 2023. [En línea]. Disponible: https://www.genome.gov/es/genetics-glossary/Codon
3. “R2 (R cuadrado) o Coeficiente de Determinación”. Estrategias de Inversión — Tu portal para invertir en Bolsa. Accedido el 19 de octubre de 2023. [En línea]. Disponible: https://www.estrategiasdeinversion.com/herramientas/diccionario/fondos/r2-r-cuadrado-o-coeficiente-de-determinacion-t-1163
4. “Máquinas de Soporte Vectorial, Clasificador Naive Bayes y Algoritmos Genéticos para la Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos”. SciELO - Scientific electronic library online. Accedido el 19 de octubre de 2023. [En línea]. Disponible: https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642018000600153#:~:text=El%20algoritmo%20clasificador%20Na%C3%AFve%2DBayes,los%20conjuntos%20de%20datos%20verdaderos.
5. “¿Qué es el algoritmo de k vecinos más cercanos? — IBM”. IBM in Deutschland, Österreich und der Schweiz — IBM. Accedido el 19 de octubre de 2023. [En línea]. Disponible: https://www.ibm.com/es-es/topics/knn