



JULIAN PRADA ORTEGA  
2212925

INTELIGENCIA ARTIFICIAL - F1

# LOL STATISTICS

Modelo de clasificación enfocado en el popular videojuego league of legends



HOME

# ¿CUÁL ES EL OBJETIVO?

Haciendo uso de un dataset dónde se recopila información acerca de todos los campeones incluidos en el juego al día de hoy, primero de junio del 2024, se intentará plantear un modelo de clasificación.





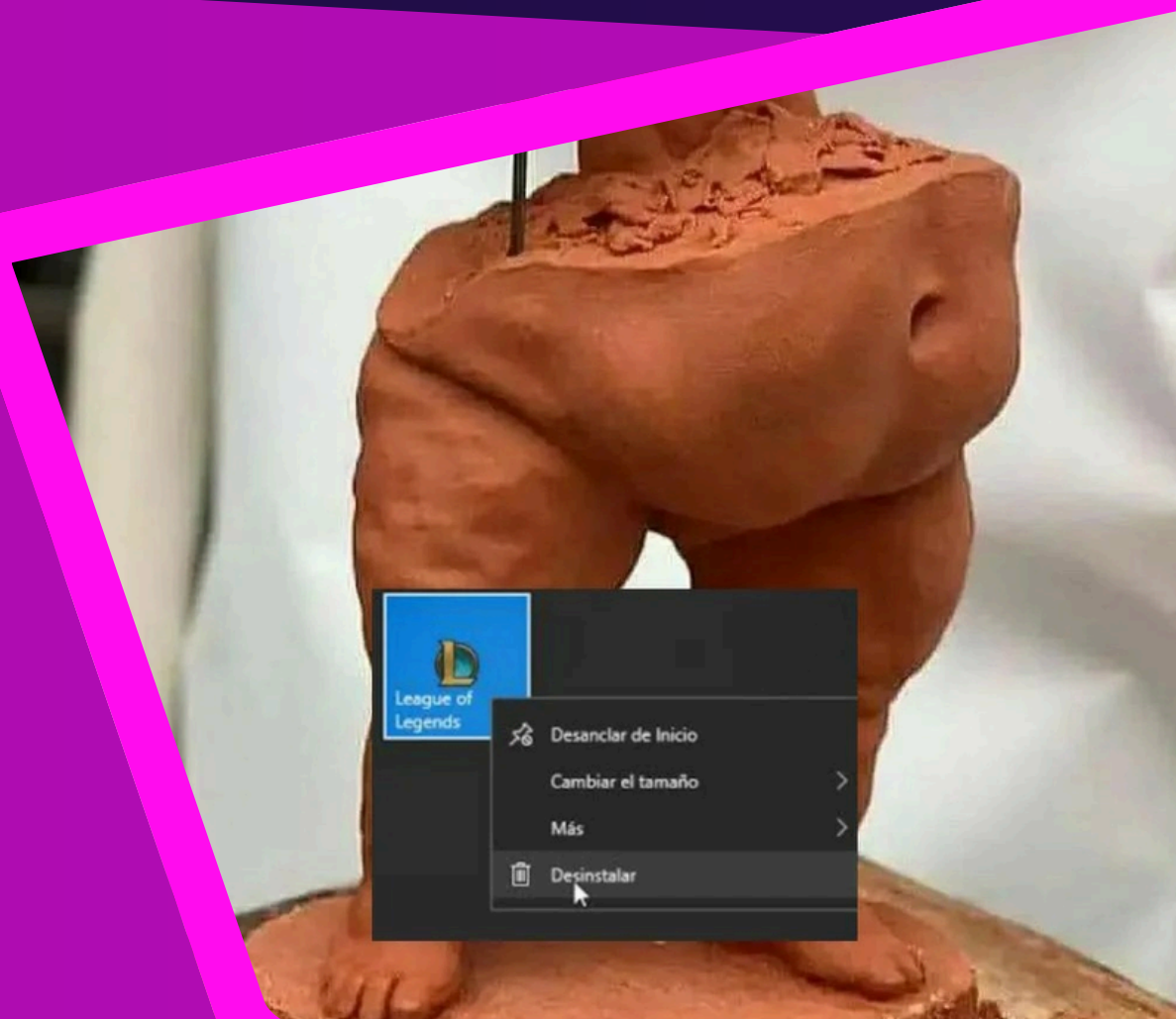


# METODOLOGÍA

Haciendo uso de una receta que consta de:

- **Recopilación de datos**, en este caso se usó un dataset que ya estaba hecho y se encuentra alojado en la plataforma de Kaggle ([aquí](#)).
- **Preprocesamiento**, dónde se hace una limpieza al dataset de posibles valores no especificados, campos vacíos, columnas del mismo que no sean de tipo numéricas, etc.
- **extracción de características**, dónde se pone evidencia, características importantes del dataset, su número de filas, columnas, se deciden las posibles columnas para trabajar, la que se va a usar par ser el ground truth, y demás.
- **algoritmo de aprendizaje**, dónde haciendo uso de modelos de aprendizaje automático se programará un algoritmo de estos para el dataset en concreto.
- y por último se **pondrá aprueba** para hacer un **procesamiento posterior** para ver como le fue al algoritmo

Todo esto con la ayuda de Python y la infinidad de librería que los respaldan.





# EXPLORANDO EL DATASET

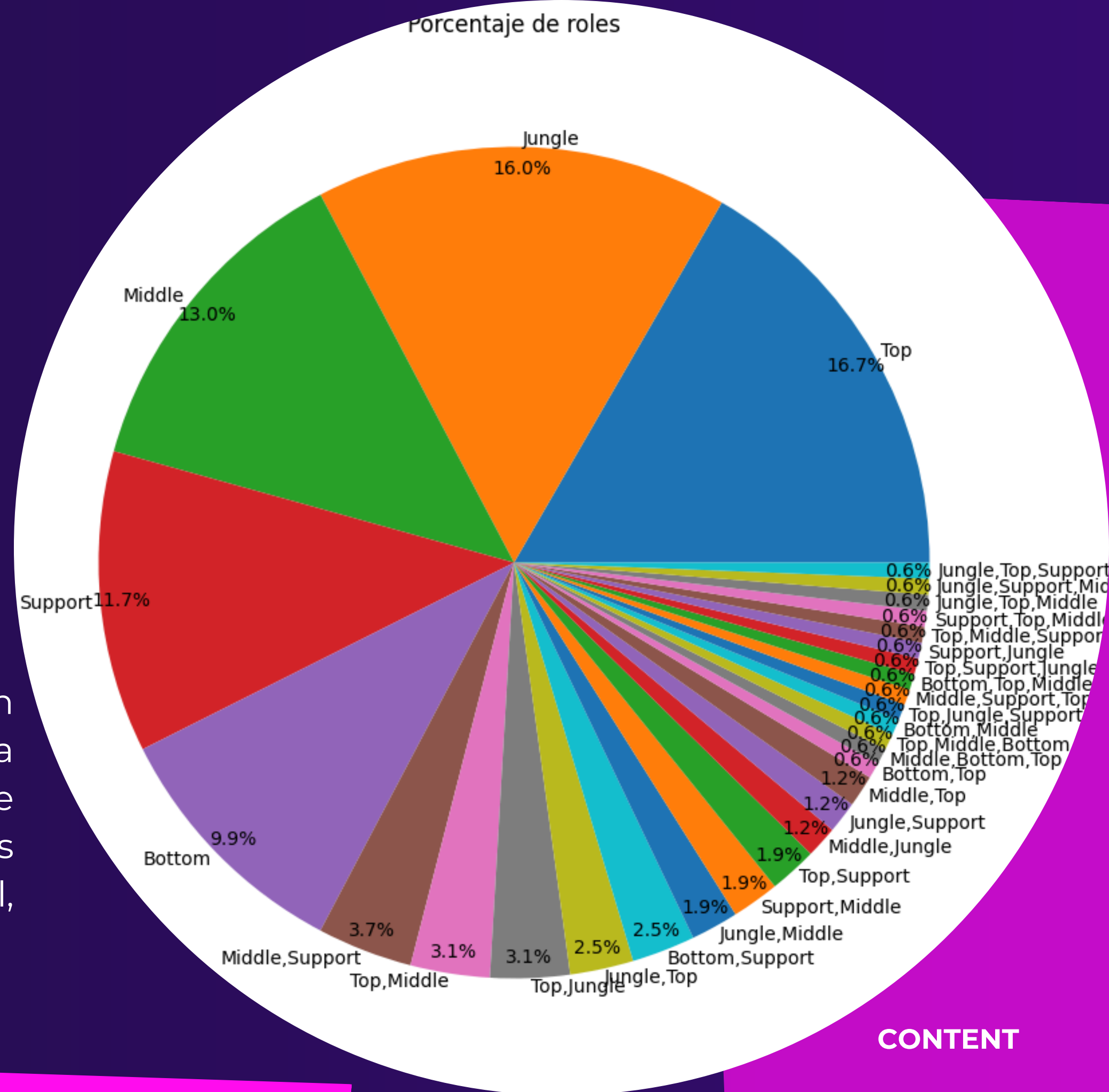
El dataset que se usará en esta ocasión cuenta con 166 filas, que representa los campeones, y 26 columnas dónde se detallan las estadísticas base de cada uno de ellos. Por fines de espacio no se detallarán muchas de estas variables, pero se puede observar con más detalle en el colab del proyecto ([aquí](#))



LOL STATISTICS

# PORCENTAJE DE ROLES

En el league of legends los campeones tienen un rol específico de cada una de las partes del mapa en dónde se debe jugar, importante destacar que el dataset discrimina por roles compartidos, es decir, campeones que poseen más de un rol oficial, son tomados como un rol aparte.

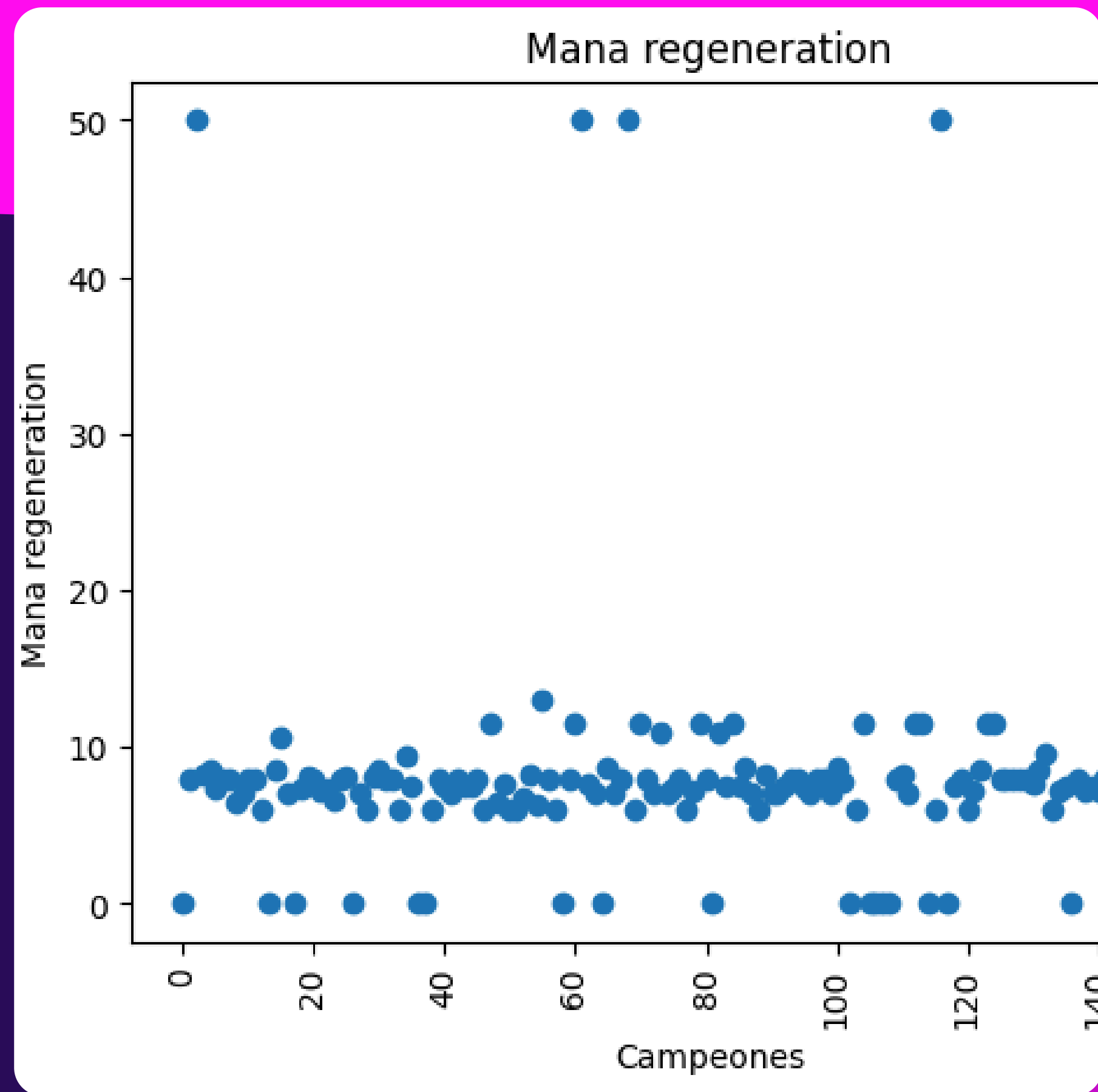


CONTENT



# REGENERCIÓN DE MANÁ

Cada campeón tiene una regeneración de maná distinta, muchas veces dependiendo del estilo de juego del campeón, como también hay otros que no hacen uso de este recurso, para aquellos casos se toma como regeneración de maná 0.

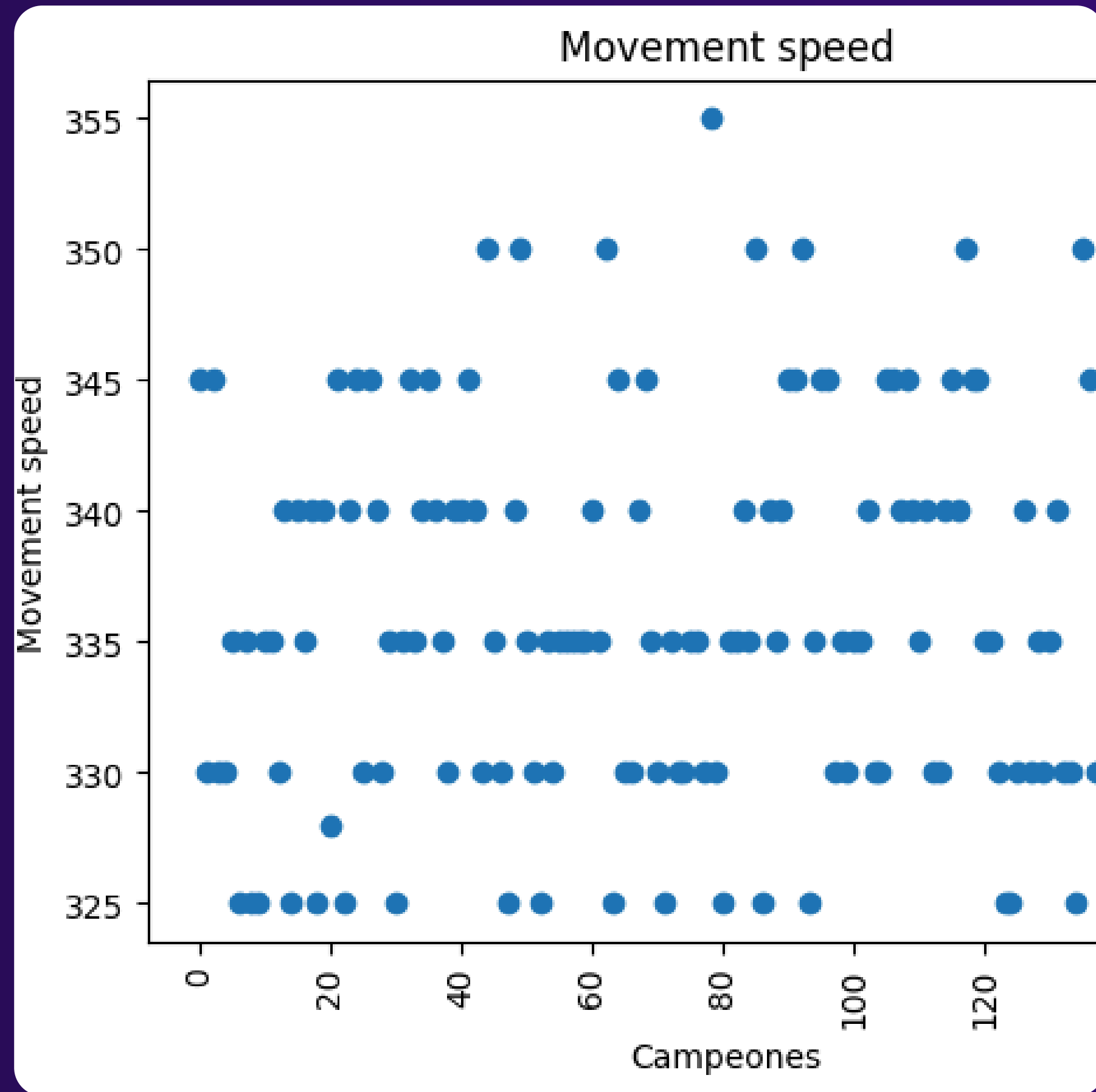




# VELOCIDAD DE MOVIMIENTO

En la velocidad se nota algo interesante, pareciera que hubiera rangos fijos, de manera que la clasificación aquí es muy sencilla, esta es una de las razones por la cual se optó por clasificar.

Y así sucede con demás variables con las que cuenta en el dataset y se ahonda más en el Google colab mencionado anteriormente.

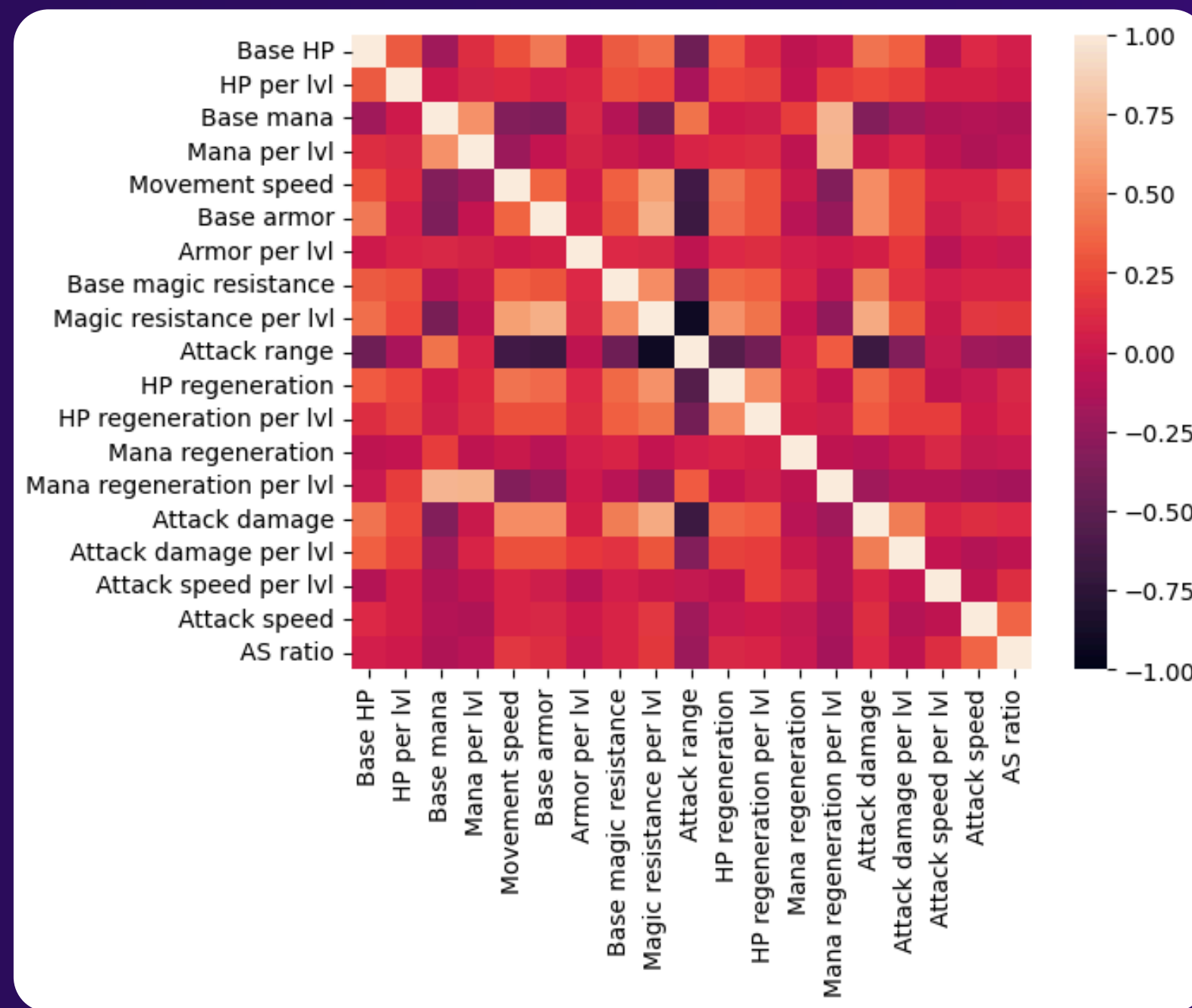






# MAPA DE CALOR

Este es el mapa de calor que muestra la relación que tiene cada una de las variables con las demás.







# ESTANDARIZACIÓN

Para hacer uso del data set y poder usar todos los modelos y técnicas es recomendable que los datos sigan una distribución normal, esto lo conseguimos estandarizando los datos, ya sea haciendo uso del método que trae la librería scikit-learn, o como en este caso aplicando una sencilla fórmula a todos los datos del dataset.

$$Z = \frac{x - \mu}{\sigma}$$

# GROUND TRUTH

Una vez después de haber hecho todo esto, siguiendo los pasos de la receta, elegiremos la columna que funcionará como nuestro **Ground truth**, es decir, el valor con el que el modelo se comparará para saber que tan efectivo es.

Al principio se tentó en hacer uso de las 4 columnas posibles que tenía el dataset para hacer clasificación, sumándolas todas, pero al final se decidió que la columna que se usaría fue la que lleva como nombre “Range type” pues los accuracy que arrojaba era muy bueno.

Range type
Melee
Ranged
Melee
Ranged
Melee
...
Melee
Melee
Ranged
Melee
Melee



# MODELOS

Pasando al apartado de los modelos, se hizo uso de 3

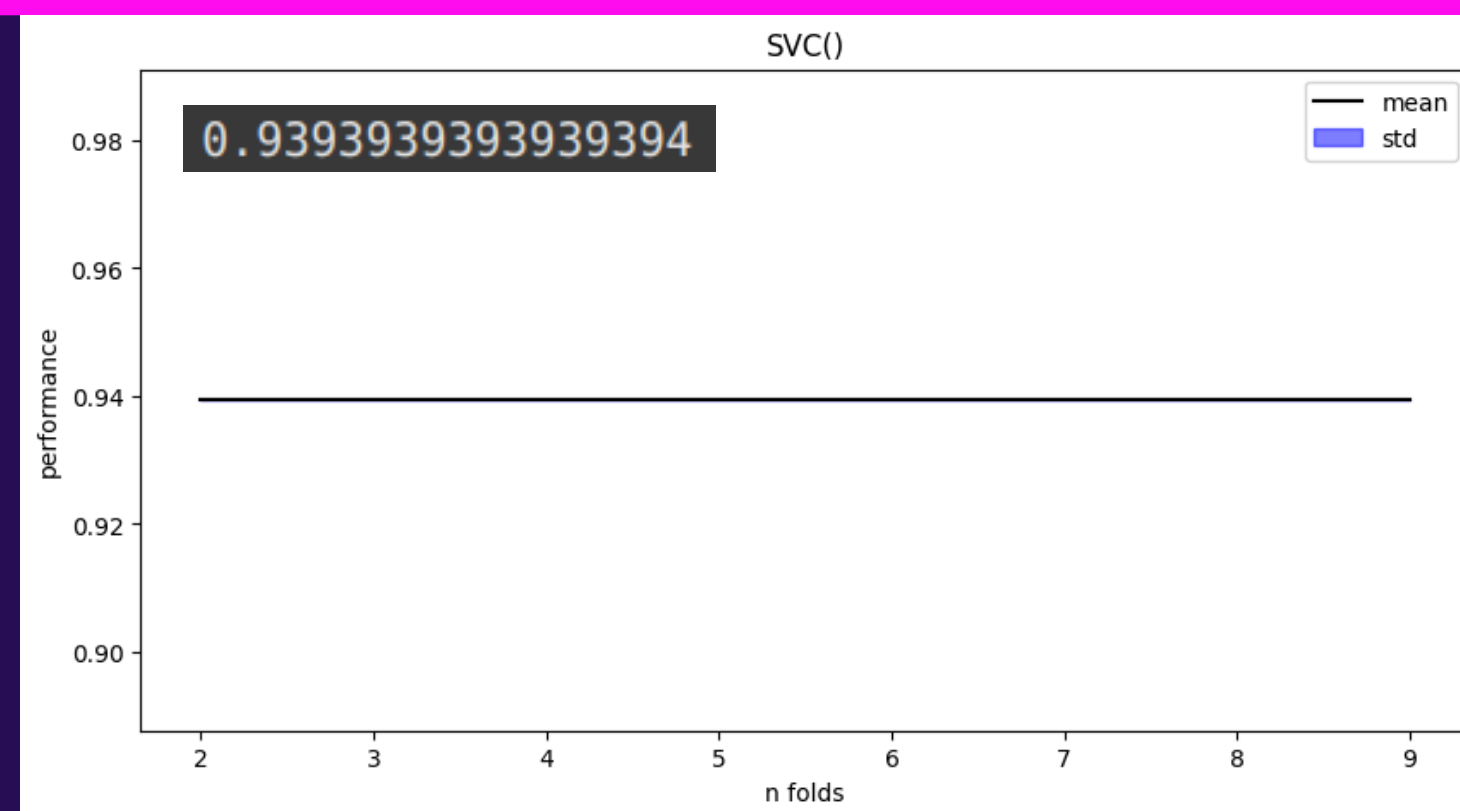
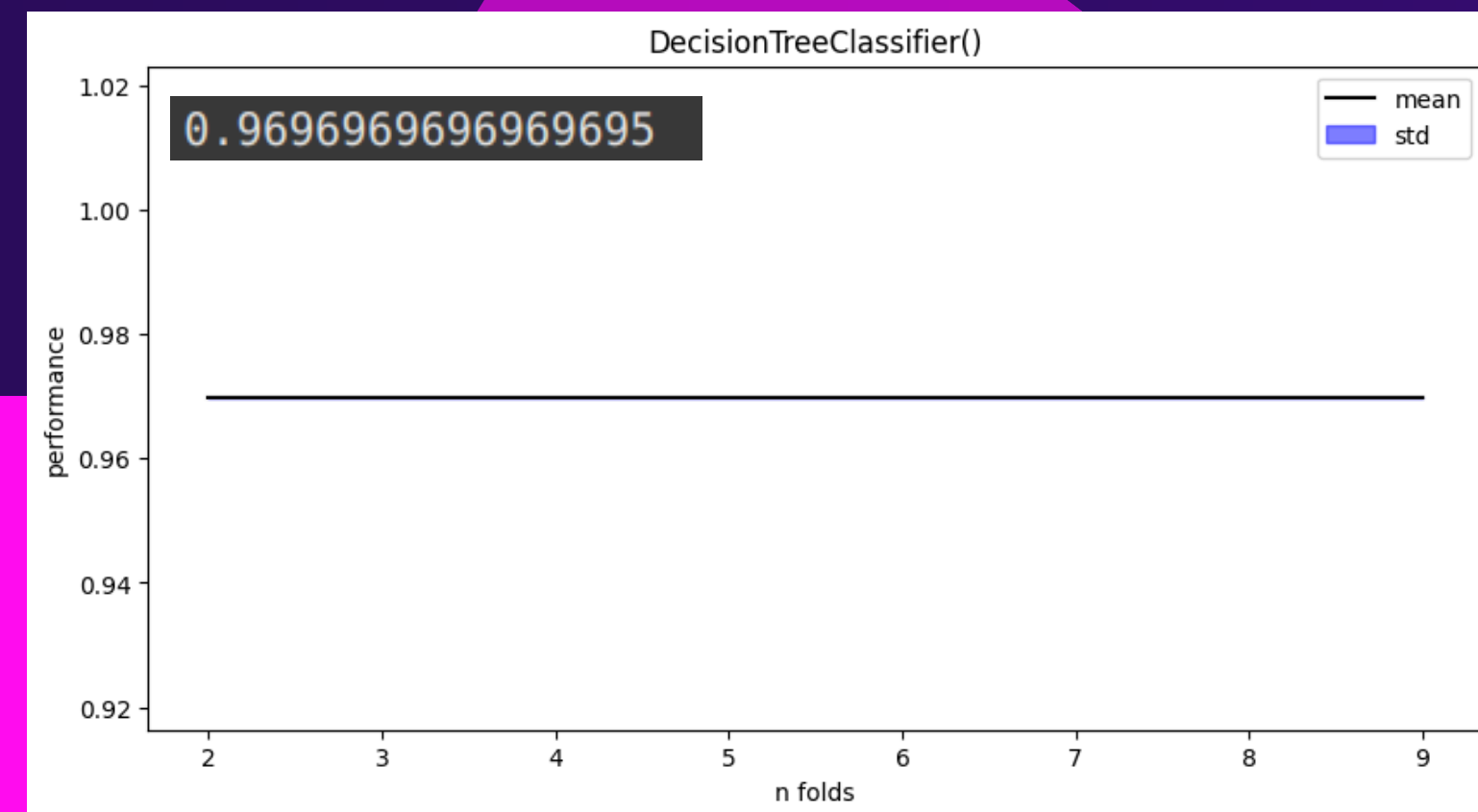
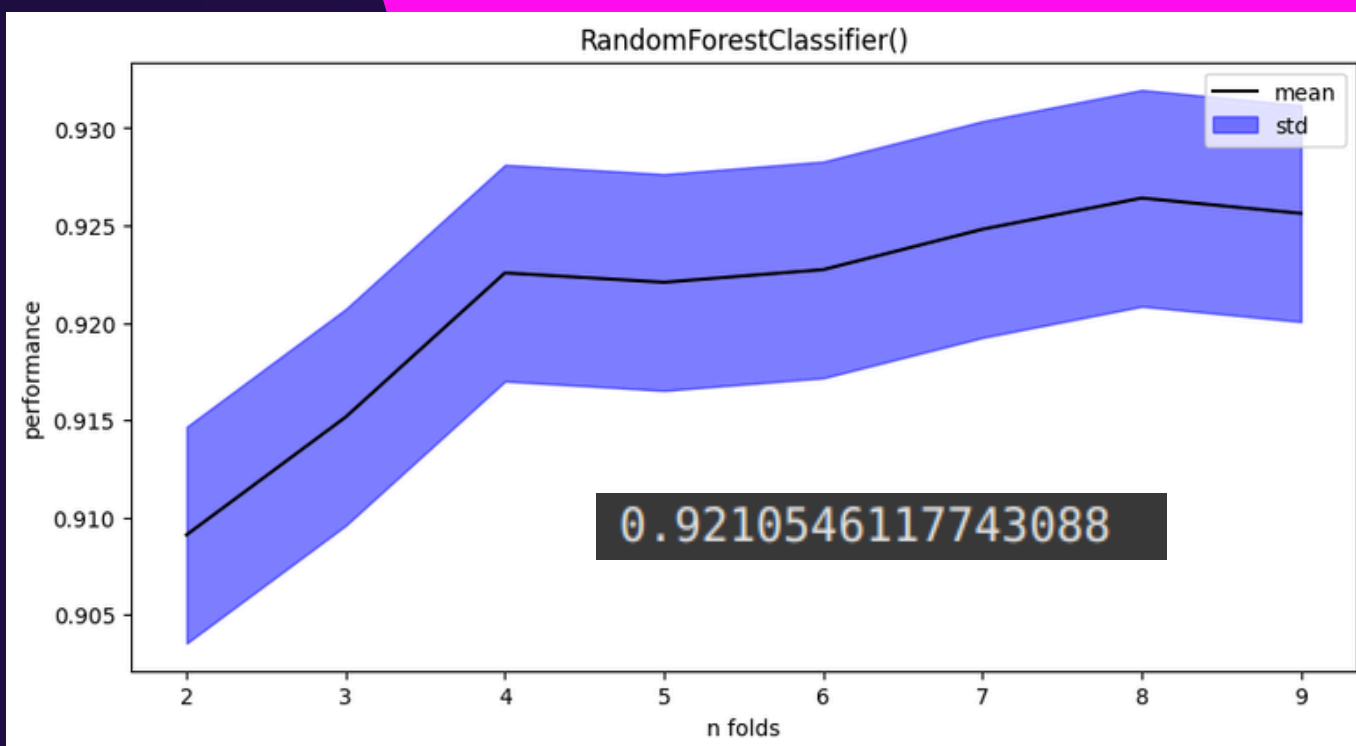
- RandomForestClassifier
- DecisionTreeClassifier
- SVC

Debido a que cada uno de estos hace uso de parámetros para alterar su comportamiento y/o mejorar su accuracy, para ello se implementó una función cuya tarea es simular un **tunning de parámetros**, dónde nos dice que parámetros son los adecuados para cada tipo de modelo; Con esto llegamos al último paso de la receta dónde graficaremos cada uno de los resultados a ver cuál es el mejor, para esto último se usaron las técnicas de “Train\_Test\_Split” y “Cross-Validation”, a continuación se muestran los resultados.



LOL STATISTICS

# TRAIN\_TEST\_SPLIT



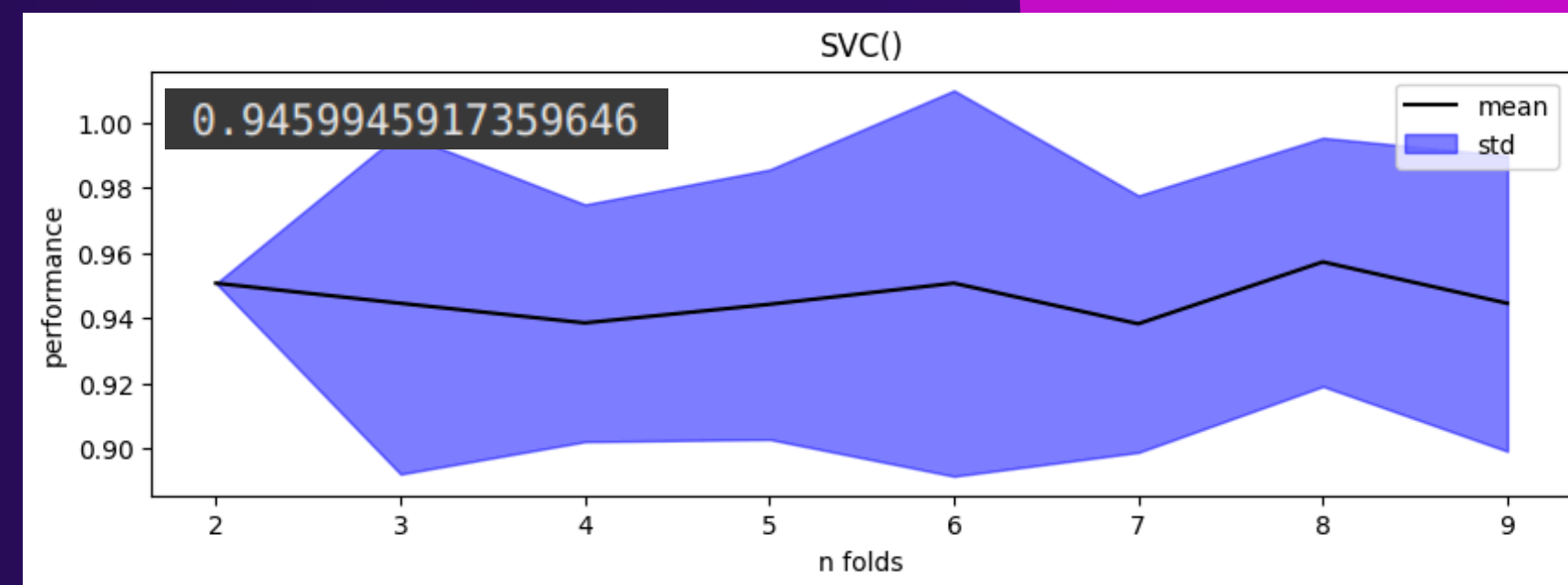
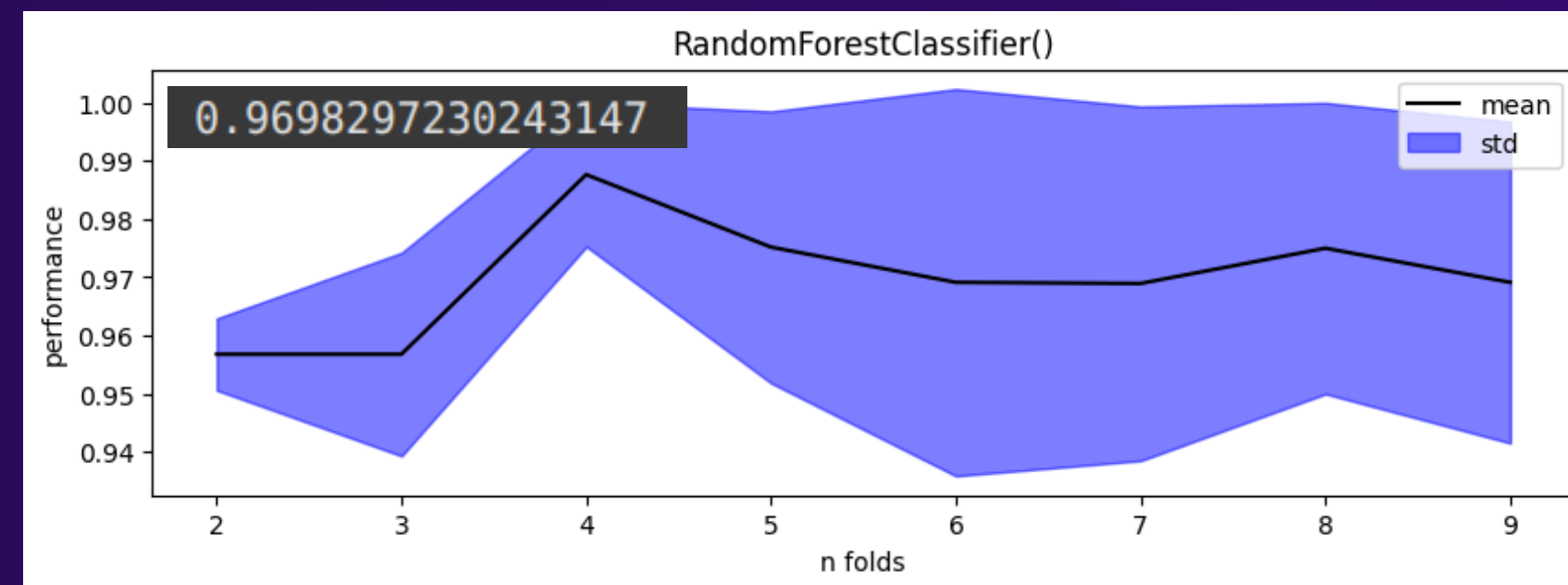
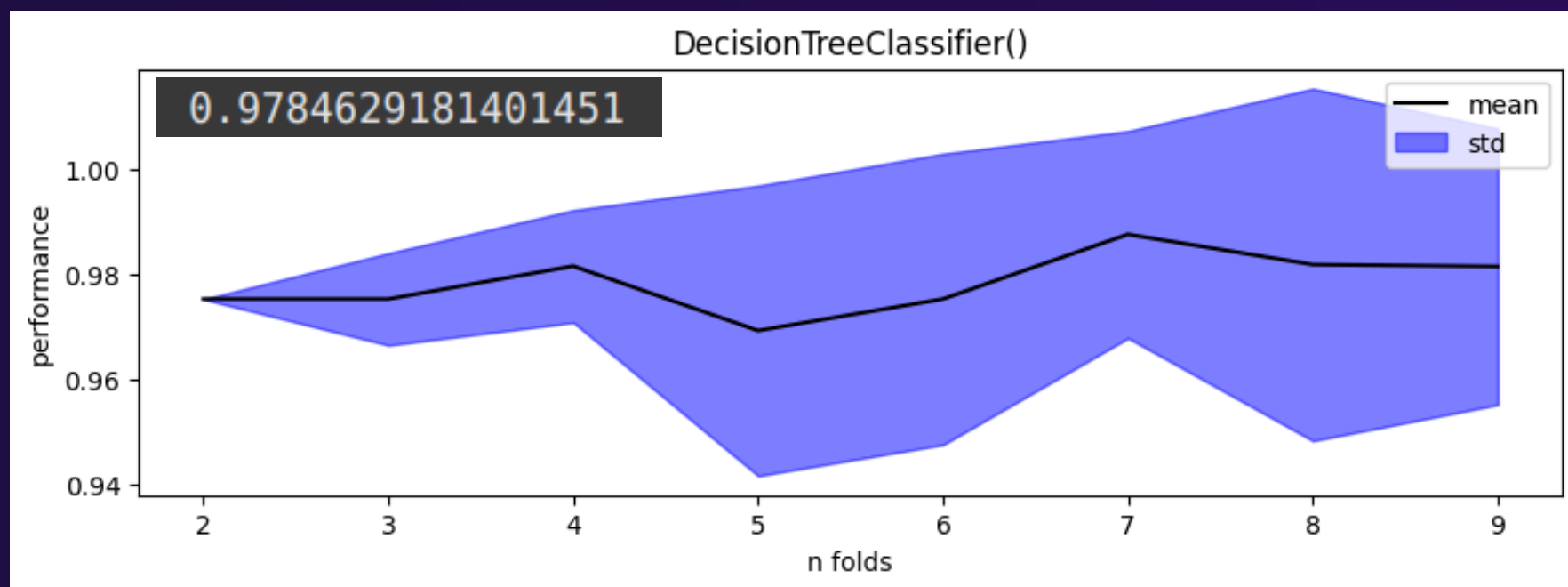
CONTENT





LOL STATISTICS

# CROSS-VALIDATION



CONTENT



# RESULTADOS

Al observar los resultados y después de las pruebas del tuning de parámetros, ambas técnicas concluyen que el mejor modelo a usar es el DecisionTreeClassifier, en el caso del train\_test\_split con parámetros como: criterion = 'entropy' que a pesar de que es más caro computacionalmente hablando es más preciso y el max\_depth en este caso donde no hay una desviación no lo dice la función.



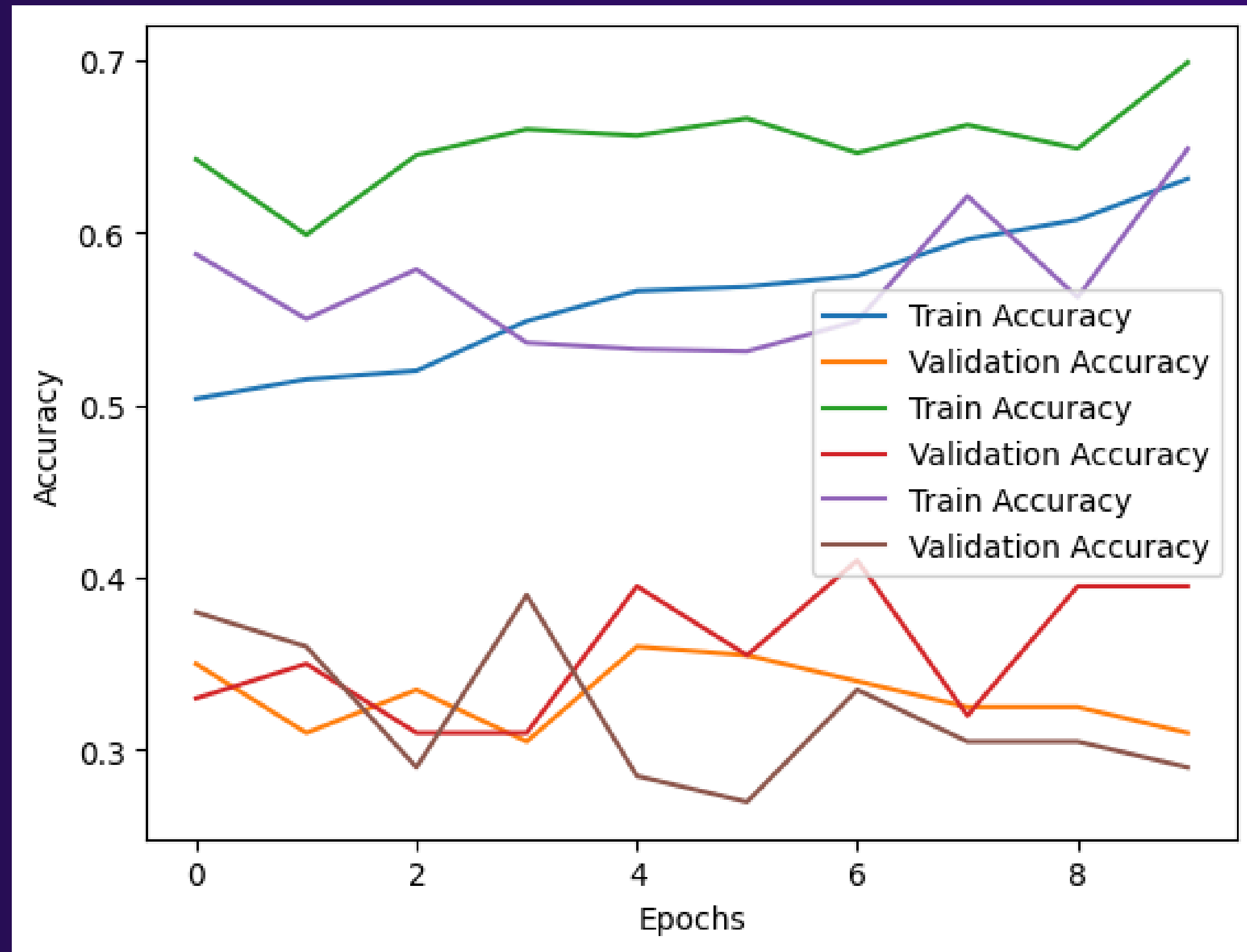
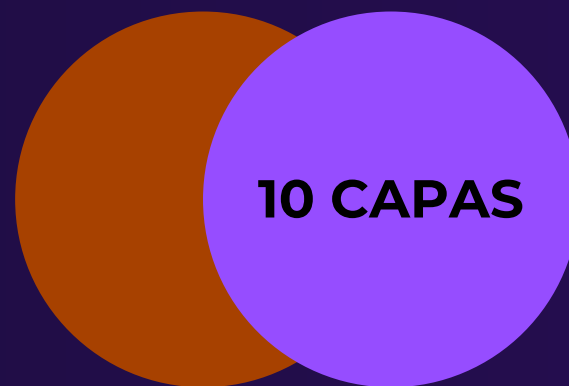
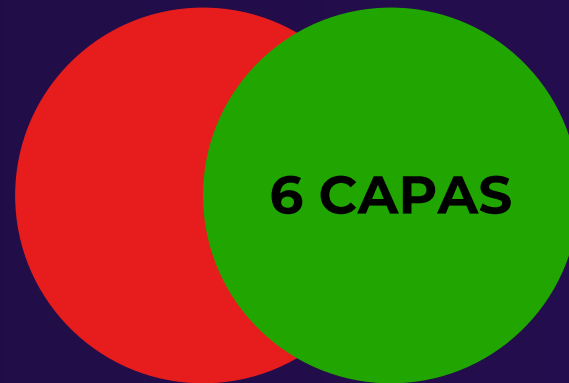
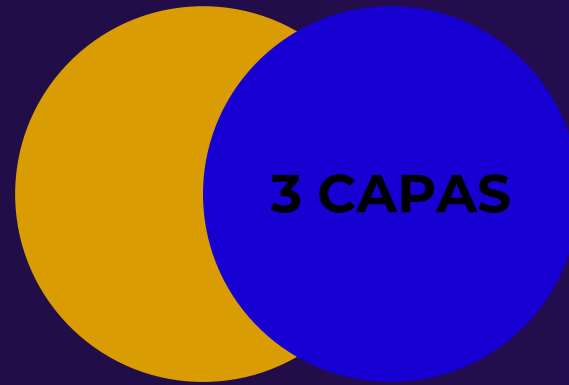
# REDES NEURONALES

Una vez sabiendo todo esto, en el contexto de machine learning, las redes neuronales son un modelo muy famoso por su poder, que intenta simular un cerebro humano; Consta de capas de neuronas artificiales, las cuales en conjunto conforman la red, están diseñadas para reconocer patrones y a la vez aprender haciendo uso de todo un proceso matemático tras bambalinas. Lo usaremos para nuestro dataset en específico.

Vamos a crear con distintas capas, con 3, 6 y 10 específicamente para ver su comportamiento.



## LOL STATISTICS







# CONCLUSIONES

¿Se logró lo esperado? Si bien el modelo de red neuronal presenta accuracy muy bajos, pues ni alterando las capas se logra obtener un valor interesante, los modelos `DecisionTreeClassifier` si convencen

¿Qué aplicaciones podría tener esto? El algoritmo de clasificación por tipo de ataque de campeón podría ser ofrecido por la misma empresa productora del juego “Riot Games” para jugadores nuevos que como curiosidad después de responder un test le diga el campeón más óptimo para él, acercando la empresa con los usuarios: Otra posible aplicación podría ser intentar predecir el nuevo campeón que se quiera sacar próximamente, a veces filtran estadísticas de los próximos campeones y con el modelo podría darse una idea del tipo que sería. Hay muchas otras más, pues los E-sport son un mercado