# Trabajo Práctico N°2
# Minería de Datos

## Tecnicatura Universitaria en Inteligencia Artificial
## FCEIA - UNR
# 2023

Integrantes
Florencia Fernández
Leandro Salvañá

**Abstract**

En este estudio se llevó a cabo la implementación de un modelo de regresión basado en árboles de decisión con el propósito de predecir las ganancias de una empresa a partir de sus gastos en investigación y desarrollo, tareas administrativas, actividades de marketing y promoción, costos generales y su ubicación geográfica. El resultado obtenido consistió en un modelo con un R^2 ajustado de 0.99, lo que indica una capacidad casi perfecta para explicar la variabilidad de la variable objetivo utilizando los predictores escogidos.

Posteriormente, se emplearon tres modelos de clasificación diferentes para predecir la clasificación de los tipos de Pokémon basándose en sus estadísticas base. Los métodos utilizados incluyeron árboles de decisión, Multinomial Naive Bayes y K-Nearest Neighbors (K-NN). La evaluación de estos modelos entrenados reveló que K-NN mostró un rendimiento superior en esta tarea de clasificación. No obstante, es importante destacar que, a pesar de esto, las métricas obtenidas fueron subóptimas, lo que lleva a la conclusión de que las estadísticas base no son predictores confiables para la clasificación de Pokémon.

# 1. Parte I: Regresión (Ítems 1 y 2)

El objetivo de la primera parte es predecir las ganancias de una Startup basado en los gastos en distintas ramas como ser: actividades de investigación y desarrollo, tareas administrativas y gastos generales y en actividades de marketing y promoción, y su localización. Para esto se utilizará una regresión mediante la utilización de un árbol de decisión.

## 1.1. Análisis exploratorio y procesamiento de los datos

Antes de implementar el modelo se realizará un análisis de los datos para conocer mejor el dataset que será utilizado para entrenar el modelo y se procesarán de ser necesarios para adecuarlos al modelo.

### 1.1.1. Descripción del Dataset

A continuación se detalla la información contenida en las columnas del dataset

**Features predictivas**

**R&D Spend**
Esta característica representa la cantidad de dinero que una startup gasta en actividades de investigación y desarrollo. Indica la inversión realizada en la creación de nuevos productos, la mejora de los existentes o el fortalecimiento de las capacidades tecnológicas en general.

**Administration**
Esta característica representa la cantidad de dinero que una startup gasta en tareas administrativas y gastos generales. Incluye costos asociados con el espacio de oficina, servicios públicos, salarios del personal no relacionado con la producción y otros gastos operativos generales.

**Marketing Spend**
Esta característica representa la cantidad de dinero que una startup invierte en actividades de marketing y promoción. Incluye gastos relacionados con campañas publicitarias, esfuerzos de marketing en línea, relaciones públicas y otras iniciativas destinadas a aumentar la visibilidad de la marca y atraer clientes.

**State**
Esta característica captura el estado en el que opera cada startup. Se representa como un objeto, lo que sugiere que contiene información categórica o textual

sobre la ubicación geográfica de la empresa. La información sobre el estado puede ser útil para analizar tendencias regionales o identificar posibles variaciones en los costos operativos y las ganancias en diferentes ubicaciones.

**Variable respuesta**

**Profit**
Esta característica representa la rentabilidad de cada startup. Indica la ganancia o pérdida financiera generada por la empresa en un período específico. La ganancia se mide típicamente restando los ingresos obtenidos de los costos operativos totales incurridos, que incluyen el gasto en investigación y desarrollo, gastos administrativos, gastos de marketing y otros factores relevantes.

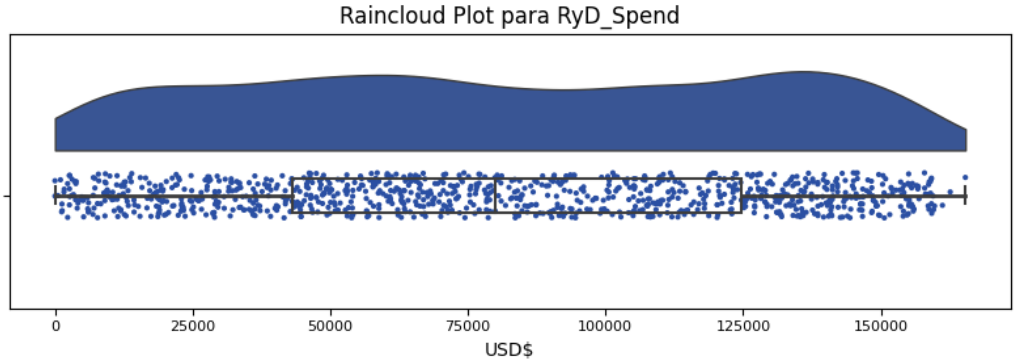
**Datos técnicos**
El Data Frame tiene un índice que es un rango que va desde 0 hasta 999, lo que significa que hay 1000 filas en el Data Frame, tiene un total de 5 columnas, no falta ningún dato. La mayoría de las columnas, al ser montos de dinero y en consecuencia números reales, están en formato float. Solo la columna que indica en qué estado opera cada startup está en formato object, es decir string para Pandas.

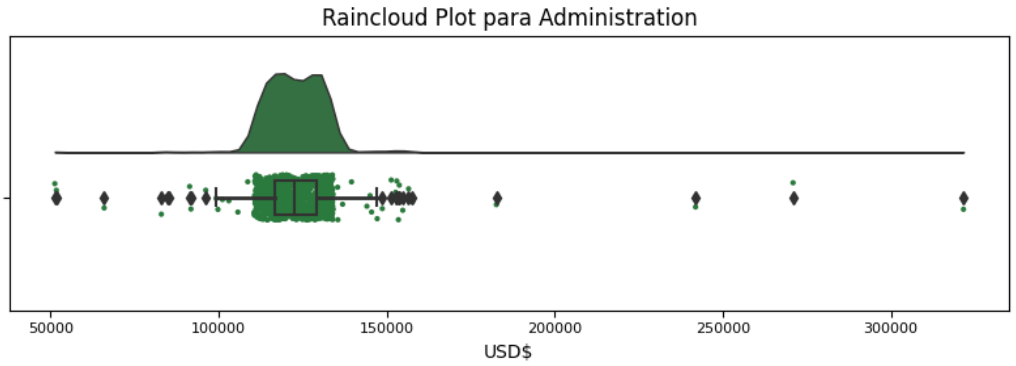
## 1.1.2. Distribuciones de los datos

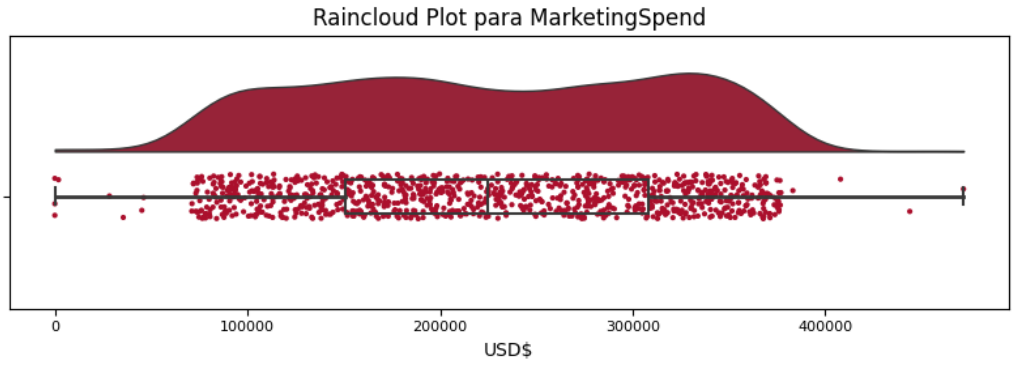
En esta sección se utilizaron raincloud plots para ilustrar la distribución de los datos.

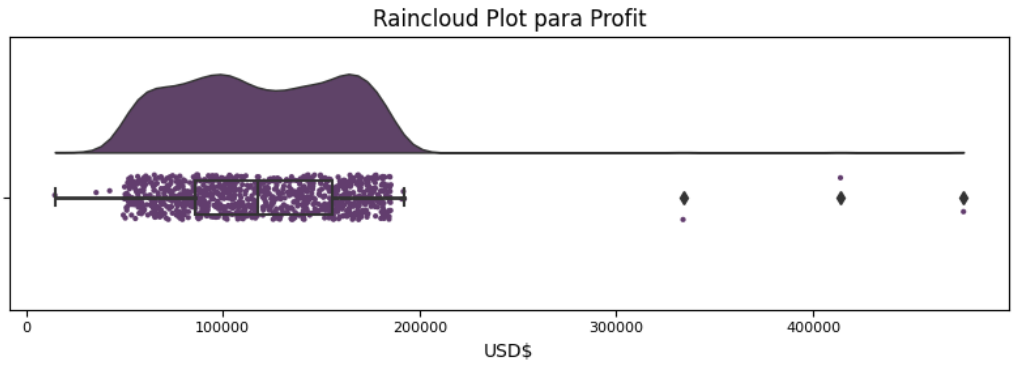
Los raincloud plots o gráficos de lluvia combinan una representación de la distribución de datos (la 'nube') con datos brutos dispersos (la 'lluvia') y se complementan aún más añadiendo boxplots.

Los resultados fueron los siguientes:





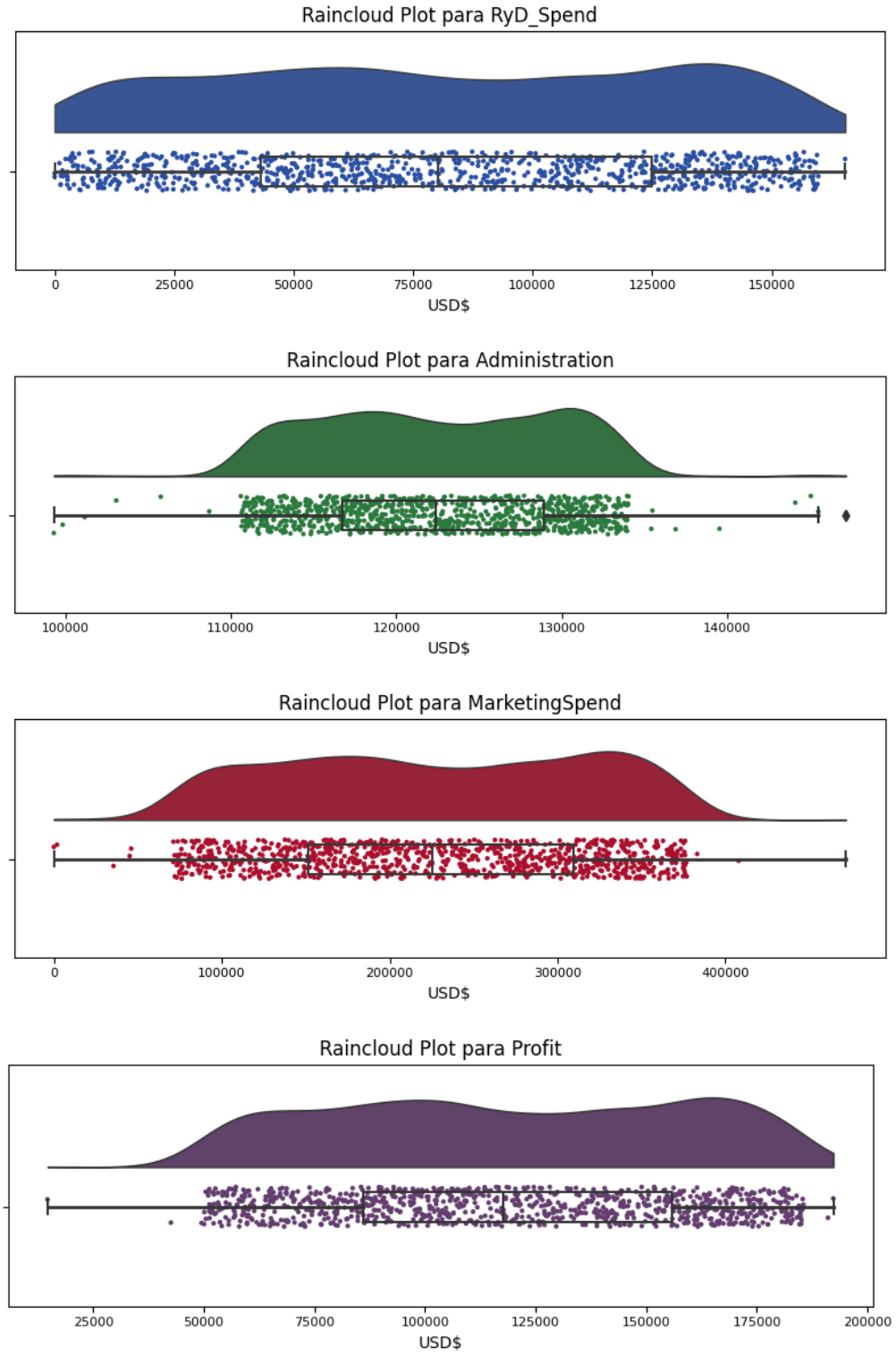

Los gráficos muestran que mientras que los montos invertidos en actividades de investigación y desarrollo y de marketing y promoción se encuentran con una distribución bastante uniforme sin una moda clara, los montos para tareas administrativas y gastos generales y ganancias, se encuentran sesgadas hacia la izquierda con unos escasos outliers hacia los valores muy altos.

## 1.1.3. Eliminación de valores atípicos

En esta sección, se procede a la identificación y eliminación de valores atípicos, mediante la selección de un factor de rango intercuartílico apropiado. El objetivo es preservar la integridad del conjunto de datos al no eliminar más del 5% de los registros, una proporción ampliamente reconocida como estándar en la industria.

Al eliminar los valores atípicos, se excluyó un 2.2% del conjunto de datos en su totalidad, lo cual se encuentra por debajo del umbral admisible universalmente del 5%.

Las distribuciones luego de la eliminación de los valores atípicos son las siguientes:



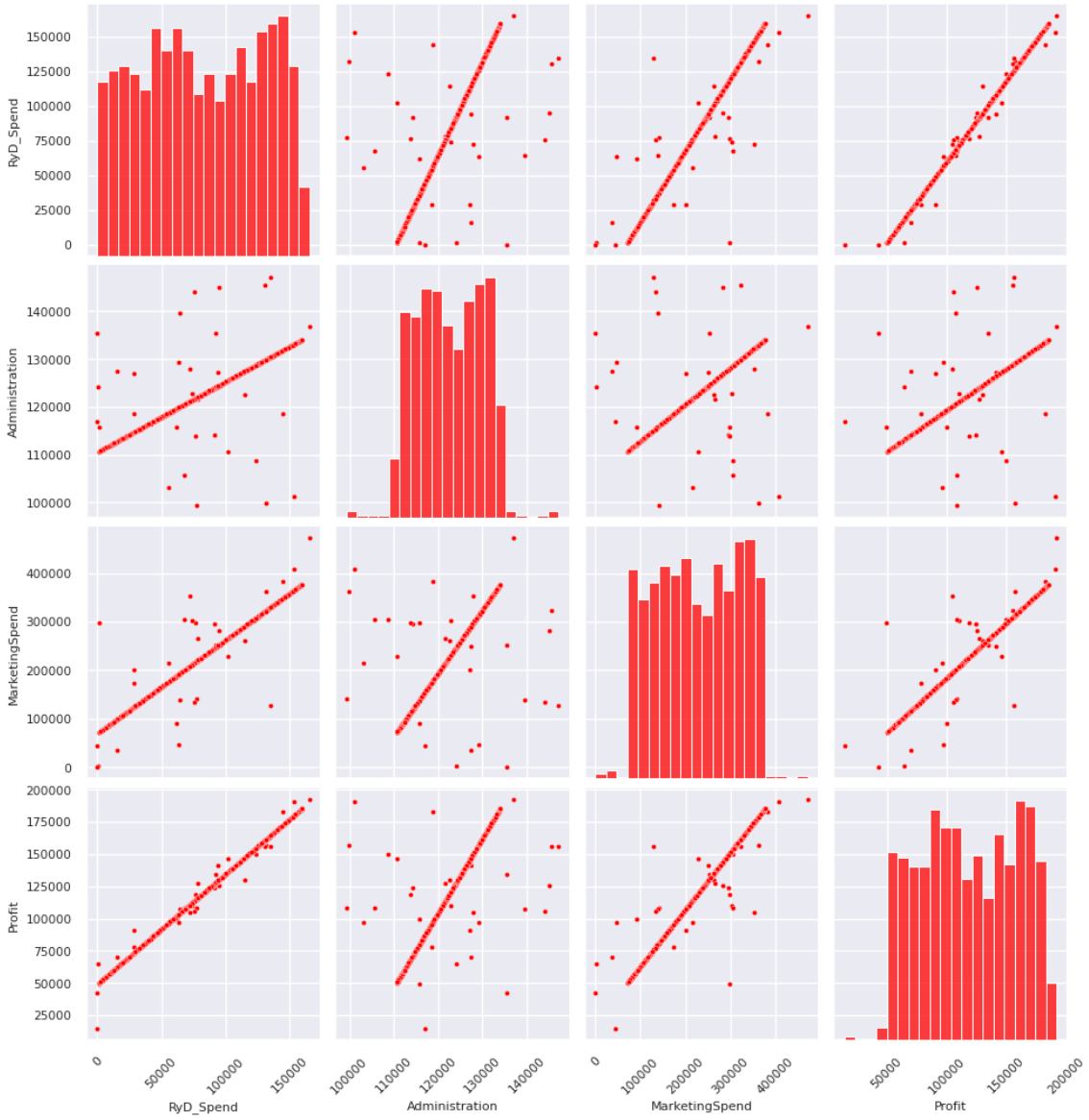
Ahora puede observarse cómo ahora todas las columnas están más uniformemente distribuídas.

## 1.1.4. Correlación entre variables

En esta sección del informe, se realiza un análisis de la correlación entre las variables presentes en el conjunto de datos. Se emplean dos enfoques principales para visualizar esta relación:

**Pairplot de Seaborn**
Se genera un conjunto de gráficos de dispersión que permite la exploración visual de la relación entre pares de variables. Este enfoque proporciona una mejor comprensión de cómo las variables interactúan entre sí y si existe una correlación lineal evidente.

El resultado es el siguiente:

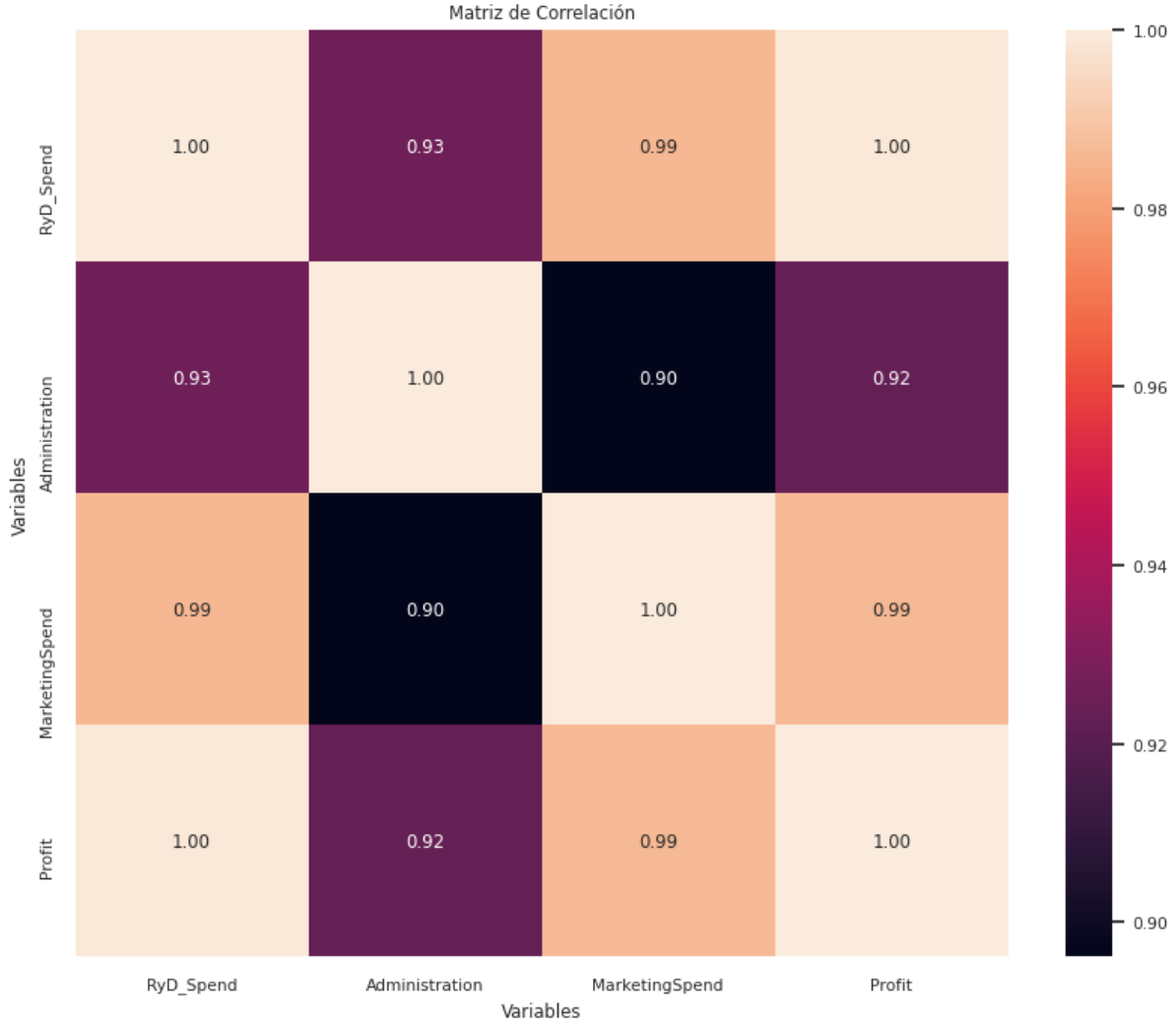


La matriz de gráficos de dispersión para los pares de características revelan una correlación lineal casi perfecta, los puntos se alinean en líneas rectas perfectamente identificables.

**Heatmap de la Matriz de Correlación**

Se crea un mapa de calor que representa la matriz de correlación de las variables. Este mapa visualiza de manera eficiente las fuerzas y direcciones de las relaciones entre las variables, destacando la intensidad de la correlación mediante una escala de colores.

El resultado es el siguiente:




Puede observarse en el mapa de calor de la matriz de correlación una alta correlación lineal entre las variables del dataset, no bajando de 0.9.

## 1.1.5. Estandarización y Codificación de variables categóricas

Se ha elegido la técnica de escalamiento Min-Max. Esta elección se basa en varios factores clave que incluyen la falta de una distribución normal en los datos, lo cual descarta la aplicación de la transformación Z-Score. Además, la

eliminación de valores atípicos previamente realizada descarta el uso de un enfoque robusto, y la inexistencia de una relación logarítmica en los datos hace inapropiado el empleo de la técnica homónima.

| | RyD_Spend | Administration | MarketingSpend | State | Profit |
|---|---|---|---|---|---|
| 0 | 1.000000 | 0.785025 | 1.000000 | New York | 192261.83000 |
| 1 | 0.927985 | 0.038905 | 0.864664 | Florida | 191050.39000 |
| 2 | 0.873136 | 0.404664 | 0.812235 | New York | 182901.99000 |
| 3 | 0.797566 | 0.011131 | 0.769126 | New York | 156991.12000 |
| 4 | 0.814128 | 1.000000 | 0.270710 | California | 156122.51000 |
| ... | ... | ... | ... | ... | ... |
| 973 | 0.635788 | 0.557236 | 0.575733 | Florida | 138841.98810 |
| 974 | 0.283025 | 0.377463 | 0.337179 | California | 89012.02672 |
| 975 | 0.587901 | 0.532833 | 0.543350 | New York | 132077.70900 |
| 976 | 0.327398 | 0.400076 | 0.367186 | California | 95279.96251 |
| 977 | 0.816272 | 0.649214 | 0.697785 | California | 164336.60550 |

978 rows × 5 columns

## 1.1.6. Codificación de variables categóricas

En esta sección se aborda el proceso de codificación de variables categóricas, un paso esencial en la preparación de los datos para su análisis y modelado. El método seleccionado para llevar a cabo esta tarea es el "one-hot encoding". Este enfoque consiste en transformar las variables categóricas en variables binarias (0 y 1) independientes, lo que facilita la interpretación por parte de los modelos de árboles de decisión para regresión.

| | RyD_Spend | Administration | MarketingSpend | Profit | State_Florida | State_New York |
|---|---|---|---|---|---|---|
| 0 | 1.000000 | 0.785025 | 1.000000 | 192261.83000 | 0.0 | 1.0 |
| 1 | 0.927985 | 0.038905 | 0.864664 | 191050.39000 | 1.0 | 0.0 |
| 2 | 0.873136 | 0.404664 | 0.812235 | 182901.99000 | 0.0 | 1.0 |
| 3 | 0.797566 | 0.011131 | 0.769126 | 156991.12000 | 0.0 | 1.0 |
| 4 | 0.814128 | 1.000000 | 0.270710 | 156122.51000 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... |
| 973 | 0.635788 | 0.557236 | 0.575733 | 138841.98810 | 1.0 | 0.0 |
| 974 | 0.283025 | 0.377463 | 0.337179 | 89012.02672 | 0.0 | 0.0 |
| 975 | 0.587901 | 0.532833 | 0.543350 | 132077.70900 | 0.0 | 1.0 |
| 976 | 0.327398 | 0.400076 | 0.367186 | 95279.96251 | 0.0 | 0.0 |
| 977 | 0.816272 | 0.649214 | 0.697785 | 164336.60550 | 0.0 | 0.0 |

978 rows × 6 columns

## 1.2. Regresión con árbol de decisión

En esta parte se separan los conjuntos de entrenamiento y de prueba, se entrena un modelo de regresión basado en árboles de decisión, se genera una representación en texto del árbol y luego visualiza el árbol de decisión.

### 1.2.1. Separar en conjuntos de entrenamiento y de prueba

En esta sección se aborda la crucial tarea de dividir el conjunto de datos en dos subconjuntos: uno destinado al entrenamiento del modelo y otro para su evaluación. Este proceso asegura que el modelo sea probado con datos no vistos durante el entrenamiento, permitiendo una evaluación imparcial de su rendimiento y capacidad de generalización.

### 1.2.2. Optimización de hiperparámetros y Entrenamiento del mejor modelo

GridSearchCV es una técnica de búsqueda de hiperparámetros que permite explorar de manera sistemática diferentes combinaciones de hiperparámetros para un modelo de machine learning, utilizando validación cruzada para evaluar la calidad de cada combinación.

Para cada combinación de hiperparámetros, GridSearchCV realiza entrenamiento y validación cruzada del modelo. Divide los datos en k conjuntos (k-folds) y utiliza k-1 conjuntos para entrenar el modelo y el conjunto restante para evaluar el rendimiento. Esto se repite k veces, de forma que cada conjunto se utiliza para la evaluación exactamente una vez.

Los hiperparámetros a optimizar son:

**max_depth**
Este hiperparámetro controla la profundidad máxima del árbol de decisión. Determina cuántas divisiones o niveles puede tener el árbol. None significa que el árbol se expandirá hasta que todas las hojas contengan un número mínimo de muestras.

**min_samples_split**
Este hiperparámetro establece el número mínimo de muestras requeridas para dividir un nodo interno. Esto significa que un nodo solo se dividirá si tiene al menos el número mínimo de muestras especificado.

**min_samples_leaf**
Este hiperparámetro establece el número mínimo de muestras requeridas para formar una hoja (nodo terminal). Esto significa que se requerirá un número mínimo de muestras en una hoja para que sea considerada válida.

**max_features**
Este hiperparámetro determina la cantidad máxima de características que se considerarán al realizar una división en un nodo.

- **auto:** Significa que todas las características se considerarán en cada división.

- **sqrt**: Significa que la raíz cuadrada del número total de características se considerará en cada división.

- **log2**: Significa que el logaritmo en base 2 del número total de características se considerará en cada división.

Los resultados fueron:

**max_depth: 10**
Cuando se establece en 10, el modelo puede realizar divisiones más profundas, lo que significa que puede considerar múltiples predictores para tomar decisiones en nodos más profundos del árbol, y por lo tanto, más específicas.

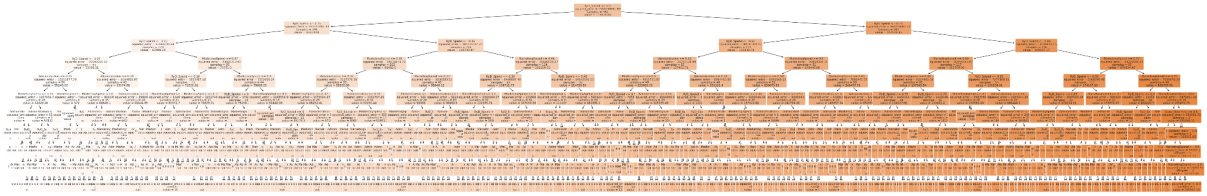
**max_features: 'auto'**
Cuando se establece en 'auto', el modelo considerará todas las características disponibles en cada nodo para tomar decisiones, lo que puede llevar a una evaluación más exhaustiva de los predictores.

**min_samples_leaf: 1**
El número mínimo de muestras requeridas en una hoja se establece en 1, lo que significa que el modelo puede crear hojas con solo una muestra. Esto permite al modelo ajustarse muy de cerca a los datos, lo que puede aumentar la importancia de las características para reducir el error en las predicciones.

**min_samples_split: 2**
El número mínimo de muestras requeridas para dividir un nodo interno se establece en 2, lo que significa que el modelo puede dividir un nodo incluso si solo hay 2 muestras disponibles. Esto permite una mayor flexibilidad en la creación de divisiones y puede destacar la importancia de las características en la toma de decisiones.

### 1.2.3 - Visualización

Esta visualización proporciona información sobre cómo el modelo toma decisiones basadas en las características de entrada.



Dado a los mejores parámetros hallados con una única instancia por hoja y que la cantidad mínima de muestras para dividir un nodo sea 2, el árbol generado es enorme.

Para observar la imagen de manera más clara se puede correr el Jupyter Notebook con el código de este trabajo, y exportará un archivo vectorial 'arbol_decision_regresion.pdf' con el gráfico.

## 1.3. Métricas

En esta sección se realiza el cálculo de diversas métricas para evaluar el rendimiento de un modelo de regresión con árboles de decisión.

Los resultados son los siguientes:

**Mean Absolute Percentage Error (MAPE)**
MAPE mide el error porcentual promedio las predicciones. En este caso, un MAPE del 2.132% es muy bajo, lo que indica que las predicciones son muy precisas y se desvían en promedio solo en un 2.132% del valor real. Esto es una muy buena señal.

**Mean Absolute Error (MAE)**
El MAE es la diferencia absoluta promedio entre las predicciones y los valores reales. Un MAE de 490.33 sugiere que, en promedio, las predicciones tienen una diferencia de aproximadamente 490.33 unidades en valor absoluto con respecto a los valores reales. Recordemos que la media de la variable respuesta 'Profit' es 119546.16, luego, el MAE representa un 0.41% de este valor

**Mean Squared Error (MSE)**
El MSE es similar al MAE, pero castiga más fuertemente los errores grandes. Un MSE de 15,471,843.003 indica que los errores cuadrados y en promedio alcanzan este valor. Cuanto más bajo sea el MSE, mejor será el modelo. Esta métrica es buena durante el proceso de elegir características, umbrales o comparar

modelos entre sí, pero no arroja muchos insights a la hora de evaluar un modelo particular.

**Root Mean Squared Error (RMSE)**
El RMSE es la raíz cuadrada del MSE y proporciona una medida de error en la misma unidad que los datos originales. Un RMSE de 3,933.426 significa que, en promedio, tus predicciones se desvían en aproximadamente 3,933.43 unidades de los valores reales, es decir, un 3.3% de la media de la viable respuesta.

**R-squared (R2)**
El coeficiente de determinación R2 es una medida de la bondad del ajuste del modelo. Un R2 de 0.99 es excelente y sugiere que el modelo explica el 99% de la variabilidad en los datos de "Profit". En otras palabras, el modelo se ajusta muy bien a los datos.

**R-squared adjusted (R2 ajustado)**
El R2 ajustado es similar al R2, pero tiene en cuenta la complejidad del modelo al penalizar el uso excesivo de predictores. Un valor de 0.99 sugiere que el modelo es sólido y no está sobreajustando.

## 1.4 - Conclusión

En este trabajo se buscó estimar las ganancias de startups utilizando un árbol de decisión de regresión. Los predictores incluyeron características clave como el gasto en investigación y desarrollo (R&D Spend), los gastos administrativos (Administration), y el gasto en actividades de marketing (Marketing Spend), así como el estado en el que opera cada startup.

El proceso de análisis comenzó con un análisis exploratorio y procesamiento de datos exhaustivo, que incluyó la identificación y eliminación de valores atípicos, la estandarización de datos y la codificación de variables categóricas. Esto garantizó la calidad de los datos y preparó el conjunto de datos para el modelado.

Se entrenó un modelo de regresión de árbol de decisión y se optimizaron los hiperparámetros utilizando grid search CV para obtener un rendimiento óptimo. Los resultados de evaluación del modelo revelaron un alto nivel de precisión, como lo demuestran las métricas obtenidas. El MAPE (Mean Absolute Percentage Error) del 2.13% indica que el modelo es altamente preciso en la estimación del profit de las startups. Además, el MAE (Mean Absolute Error) de 490.33 y el RMSE (Root Mean Squared Error) de 3933.43 sugieren que las predicciones del modelo están en línea con los valores reales, con un bajo nivel de error.

El R-cuadrado (R2) de 0.99, y el R-cuadrado ajustado (R2 ajustado) de 0.99 indican que el modelo es capaz de explicar casi la totalidad de la variabilidad en los datos de profit. Estos valores destacan la fuerte capacidad del modelo para capturar las relaciones entre las características predictoras y las ganancias de las startups.

En conclusión, este trabajo ha demostrado que el uso de un árbol de decisión de regresión junto con un análisis exhaustivo de datos y la optimización de hiperparámetros puede ser una estrategia efectiva para estimar el profit de startups basado en características clave al menos en lo que respecta a este conjunto de datos. Los resultados indican que el modelo es altamente preciso y puede ser una herramienta valiosa para la toma de decisiones empresariales al proporcionar estimaciones confiables de las ganancias esperadas en función de los gastos en investigación y desarrollo, gastos administrativos, gastos de marketing y la ubicación geográfica de la empresa.

# 2. Parte II: Clasificación (ítems de 3 a 6)

## 2.1. Análisis exploratorio y procesamiento de los datos

### 2.1.1. Conocer el dataset

En esta sección conoceremos las columnas del dataset, sus nombres, cantidad, tipo, datos faltantes y medidas de localización y centralidad para las columnas numéricas.

Cada instancia del dataset se corresponde con un pokémon. Cuenta con ocho (8) columnas, el nombre y el tipo del pokémon de tipo object (texto) y luego las seis (6) estadísticas base de los pokémon tipo int64 (enteros).

```
Información del DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1190 entries, 0 to 1189
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Name     1190 non-null   object
 1   Type     1190 non-null   object
 2   HP       1190 non-null   int64
 3   Attack   1190 non-null   int64
 4   Defense  1190 non-null   int64
 5   Sp. Atk  1190 non-null   int64
 6   Sp. Def  1190 non-null   int64
 7   Speed    1190 non-null   int64
dtypes: int64(6), object(2)
memory usage: 74.5+ KB
```

El DataFrame cuenta con un total de 1190 filas y no tiene datos faltantes.

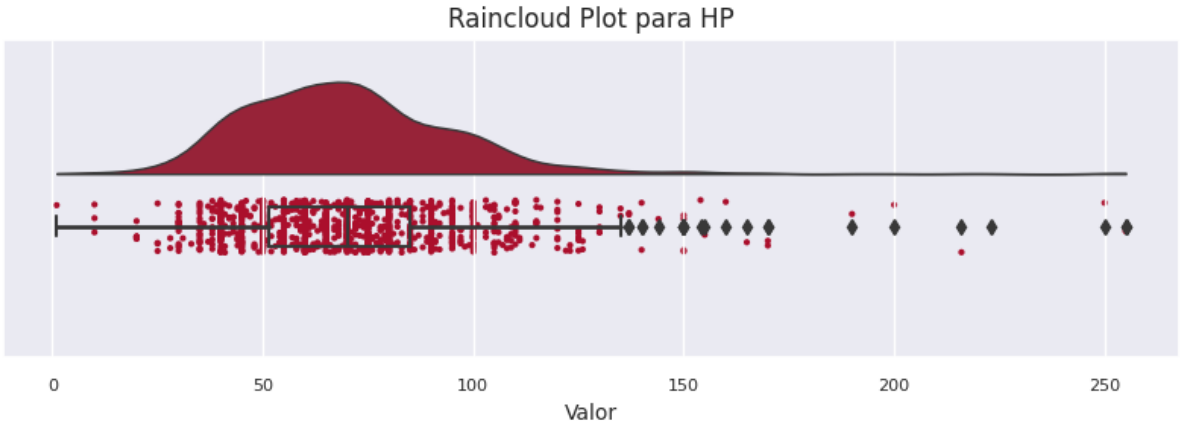
Luego se calcularon algunas estadísticas descriptivas, medidas de localización y centralidad.

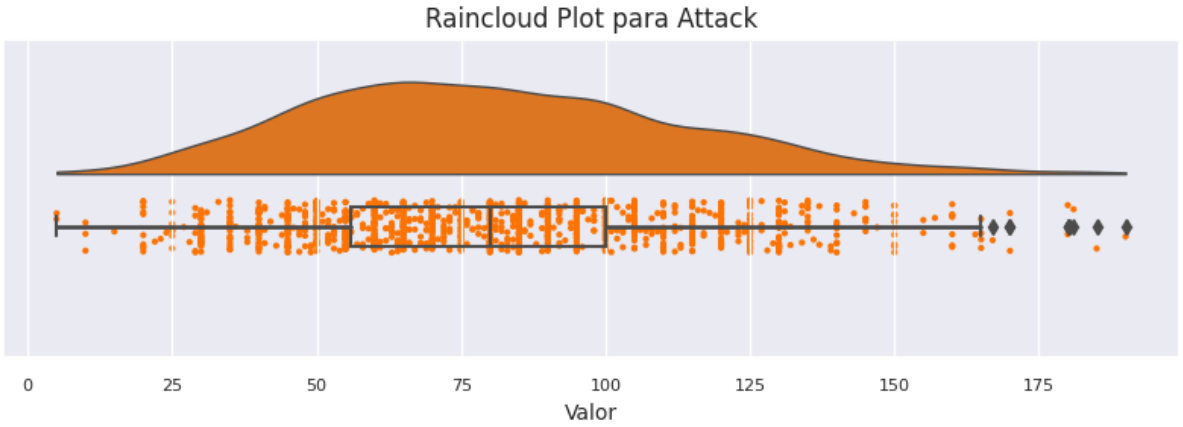
Los valores de las columnas de enteros no se salen del rango (1, 255). Lo que tiene sentido porque las estadísticas base de los Pokémon están programadas para caber en 1 byte y ser más eficientes en almacenamiento, 1 byte = 8 bits → (2^8 = 256) entonces los valores no pueden salir del rango (0, 255).

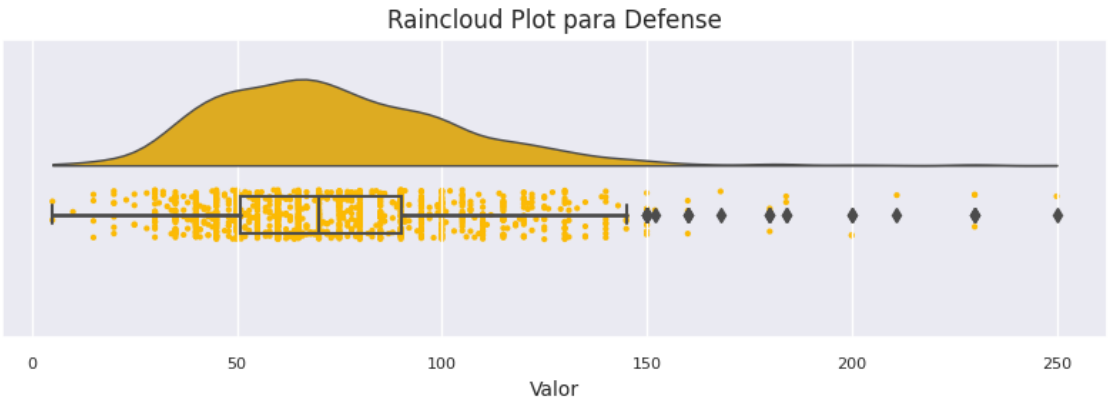
También se observa que los valores correspondientes a la media y la mediana son muy similares en casi todas las columnas por los que las distribuciones son simétricas con una leve tendencia hacia los valores más pequeños.

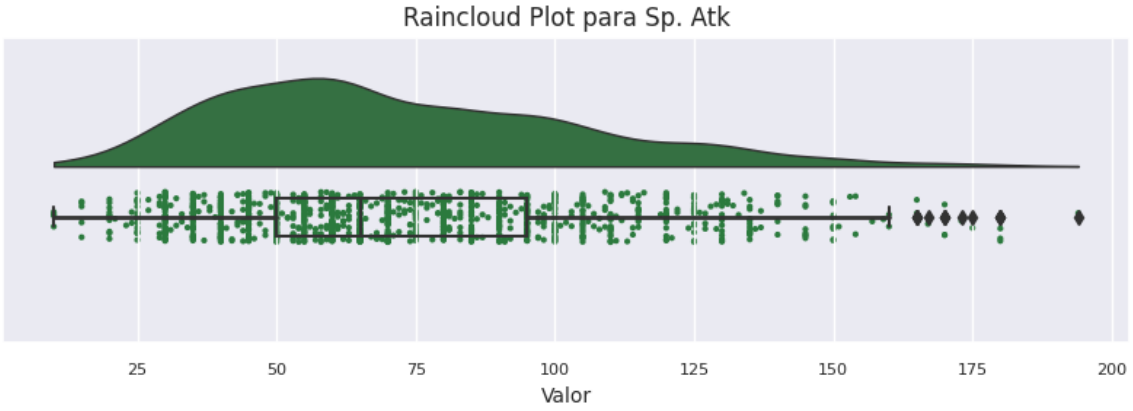
### 2.1.2. Distribuciones de los datos

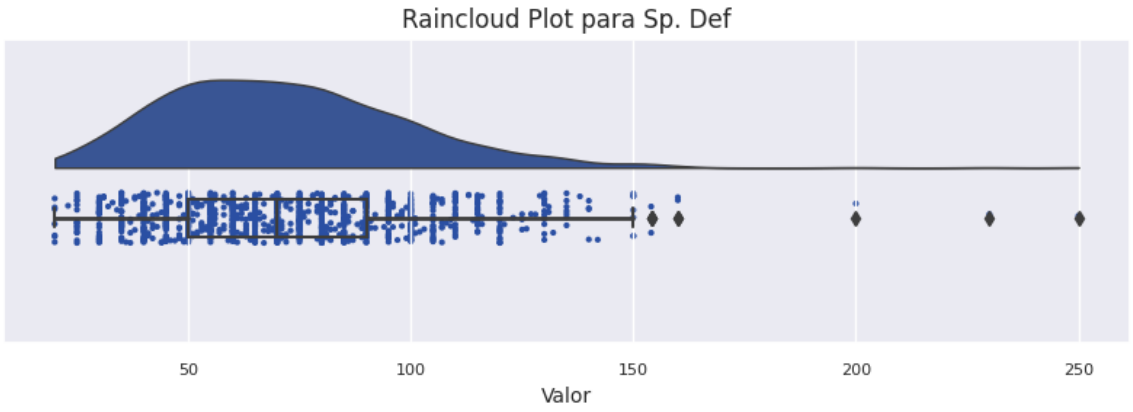
En esta sección se utilizan raincloud plots, descritos anteriormente, para ilustrar la distribución de los datos.

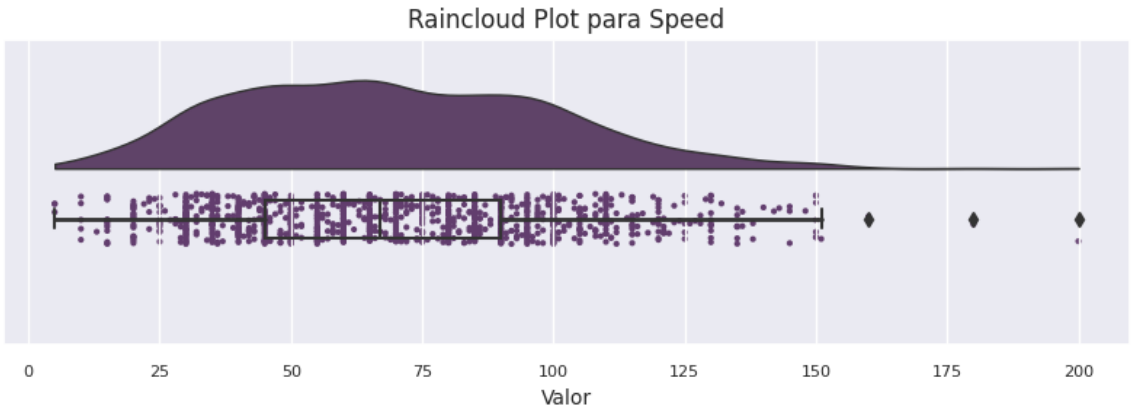



Raincloud Plot para Defense
0
50
100
150
200
250
Valor


Raincloud Plot para Sp. Atk
25
50
75
100
125
150
175
200
Valor


Raincloud Plot para Sp. Def
50
100
150
200
250
Valor


Raincloud Plot para Speed
0
25
50
75
100
125
150
175
200
Valor

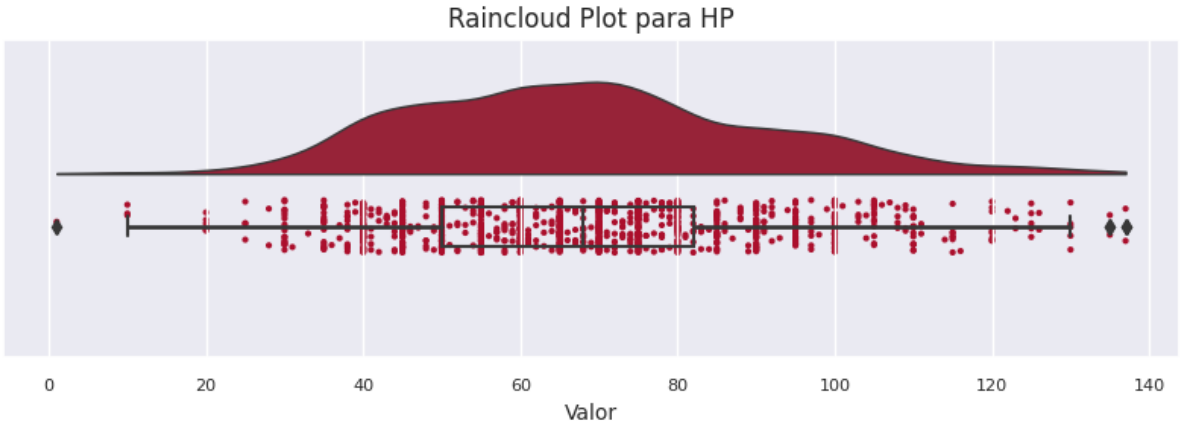
Los gráficos muestran distribuciones asimétricas positivas, es decir los datos se concentran en los valores más bajos. Conociendo el dataset, estos valores anormalmente altos pueden ser legendarios o bien pokémon como Ninjask o Shuckle que tienen valores extremadamente altos de velocidad y defensas respectivamente, pero no son representativos del tipo Bicho (la clase objetivo). Por esto último serán eliminados para el entrenamiento de los modelos.

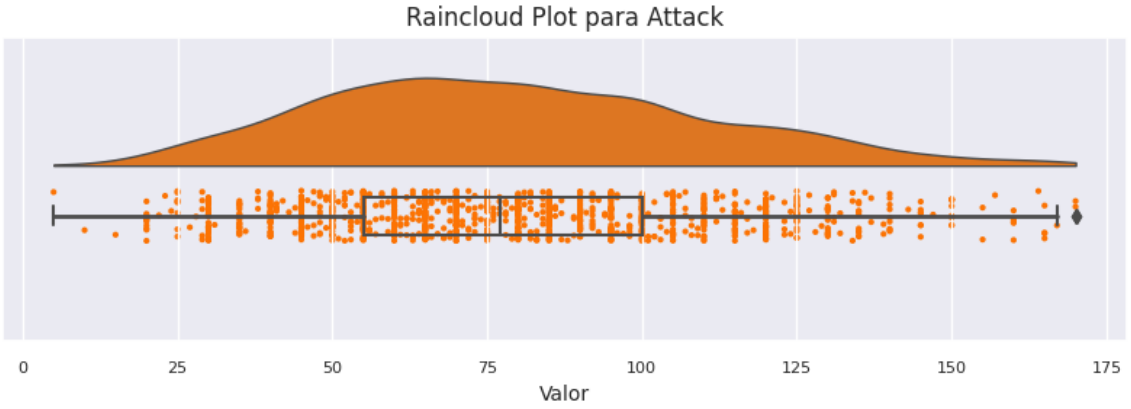
### 2.1.3. Eliminación de outliers

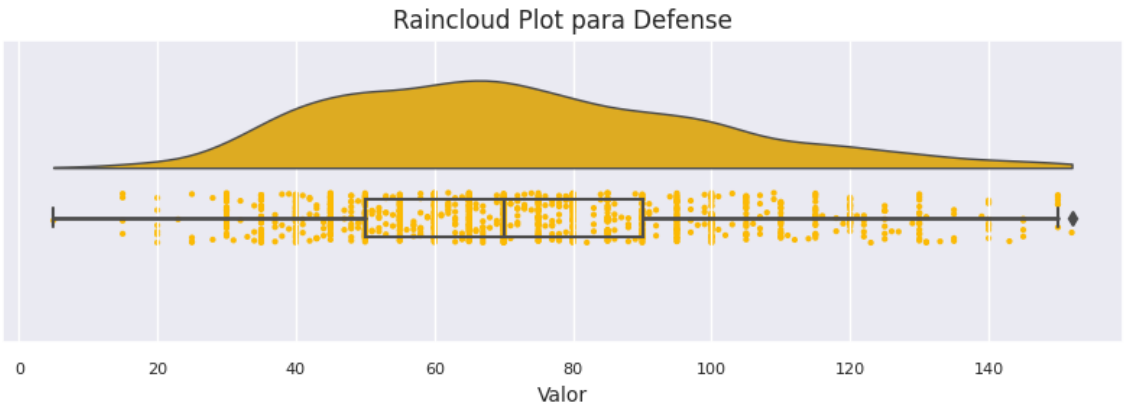
En esta sección, se procede a la identificación y eliminación de valores atípicos, mediante la selección de un factor de rango intercuartílico apropiado. El objetivo es preservar la integridad del conjunto de datos al no eliminar más del 5% de los registros.

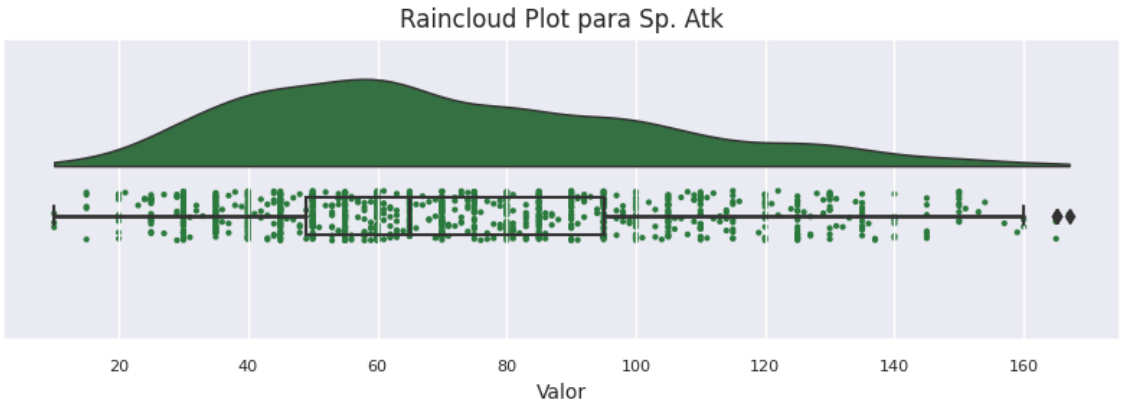
Al eliminar los valores atípicos, se excluyó un 4.5% del conjunto de datos en su totalidad, lo cual se encuentra por debajo del umbral admisible universalmente del 5%.

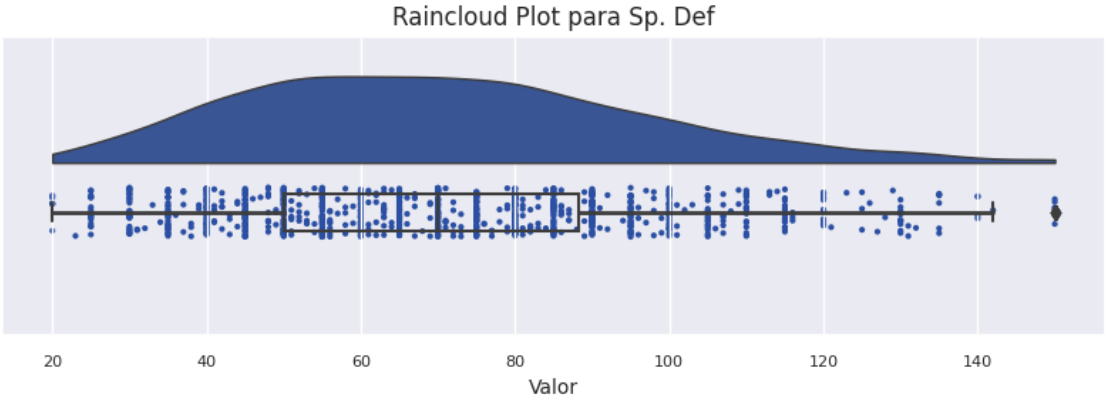
Las distribuciones de las columnas quedan, entonces, de la siguiente manera:

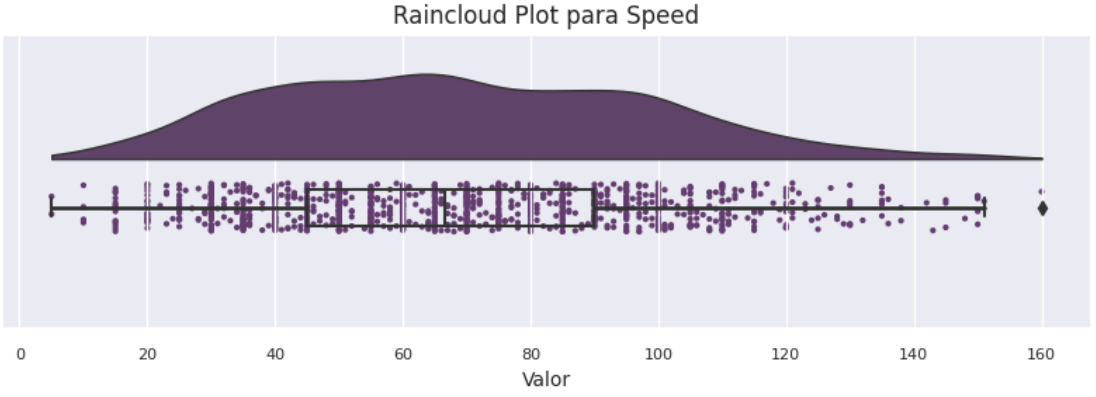



Raincloud Plot para Defense
0
20
40
60
80
100
120
140
Valor


Raincloud Plot para Sp. Atk
20
40
60
80
100
120
140
160
Valor

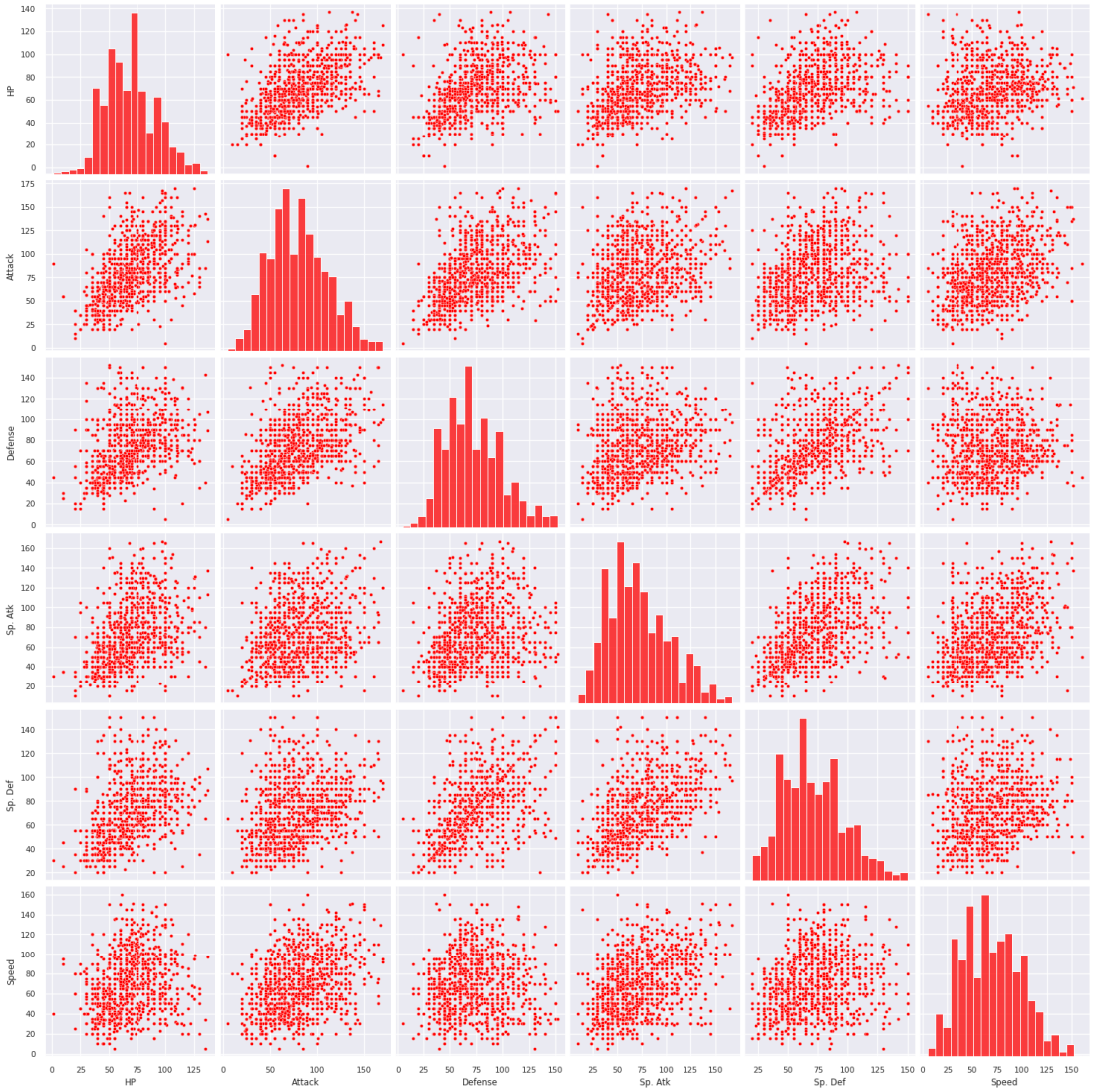

Raincloud Plot para Sp. Def
20
40
60
80
100
120
140
Valor


Raincloud Plot para Speed
0
20
40
60
80
100
120
140
160
Valor

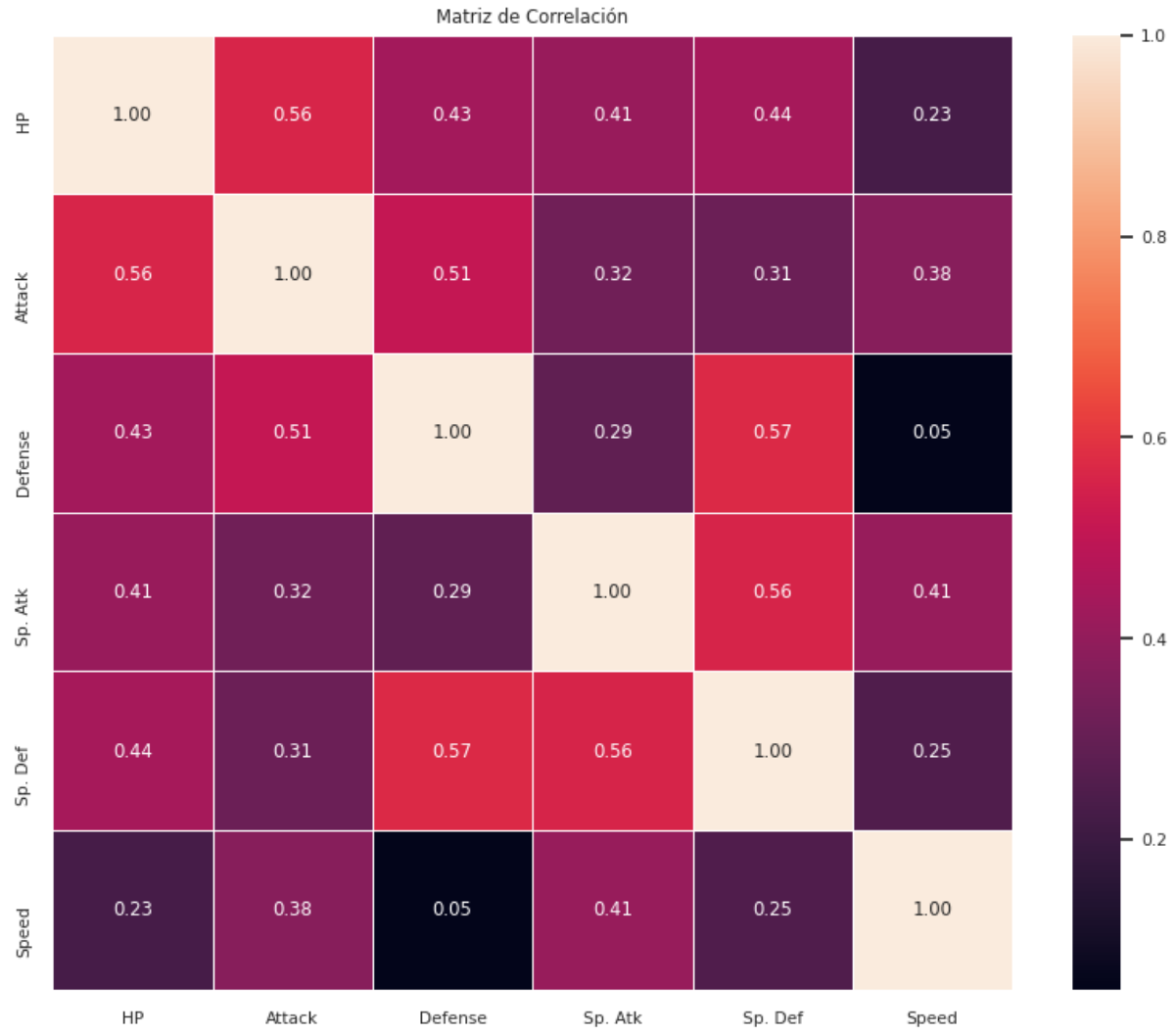
Tras la eliminación de los valores atípicos y la representación gráfica de las distribuciones, se aprecia claramente cómo todas las columnas presentan ahora una distribución más simétrica.

## 2.1.4. Correlación entre variables

En esta sección, se procede a la evaluación de las relaciones de correlación lineal entre las variables.




En esta matriz de gráficos de dispersión puede observarse que algunos pares de características presentan una relación lineal muy pronunciada, pudiendo incluso verse líneas perfectas como en los pares "Defensa - Defensa Especial" y "Ataque - Ataque Especial".




En los gráficos de dispersión y en la matriz de correlación, es evidente la presencia de correlaciones lineales significativas, como la correlación entre "Defensa" y "Defensa Especial" (0.57), así como entre "Ataque" y "Vida" (0.56), por mencionar algunas. Este hallazgo es de particular relevancia, ya que podría tener un impacto negativo en los modelos que se utilizarán, como el algoritmo Naive Bayes. Este último, se basa en la suposición de que los predictores son condicionalmente independientes, una suposición que en este caso parece no cumplirse.

## 2.2. Clasificación con árbol de decisión

En esta parte se separan los conjuntos de entrenamiento y de prueba, se entrena un modelo de clasificación basado en árboles de decisión, se genera una representación en texto del árbol y luego visualiza el árbol de decisión.

### 2.2.1. Separar en conjuntos de entrenamiento y de prueba

En esta sección se aborda la crucial tarea de dividir el conjunto de datos en dos subconjuntos: uno destinado al entrenamiento del modelo y otro para su

evaluación. Este proceso asegura que el modelo sea probado con datos no vistos durante el entrenamiento, permitiendo una evaluación imparcial de su rendimiento y capacidad de generalización.

### 2.2.2. Crear una instancia del clasificador

En esta sección, se crea una instancia del clasificador de árbol de decisión. Esta instancia se utiliza como base para construir un modelo de clasificación utilizando la técnica de árboles de decisión. El objeto representa el modelo de árbol de decisión antes de ajustar sus hiperparámetros o realizar cualquier entrenamiento con datos reales. Posteriormente, se pueden configurar los hiperparámetros y ajustar el modelo a un conjunto de datos específico para su posterior uso en la clasificación de observaciones.

### 2.2.3. Entrenar modelos y optimizar hiperparámetros

Esta sección se enfoca en la utilización de la biblioteca GridSearchCV con el propósito de identificar y ajustar de manera óptima los hiperparámetros de nuestro modelo, seleccionando las configuraciones más adecuadas a partir de un conjunto de opciones predefinidas.

El resultado fue el siguiente:

El **criterio de separación óptimo** es '**entropy**'. En los árboles de decisión, el criterio se utiliza para medir la calidad de la división de los nodos del árbol. 'Entropy' es una medida de la impureza de los nodos y se utiliza para maximizar la ganancia de información.

La **profundidad máxima del árbol de decisión** es *5*. Esto significa que el árbol se dividirá en un máximo de *5* niveles desde el nodo raíz hasta las hojas. Una profundidad máxima más pequeña puede dar como resultado un modelo más simple, mientras que una profundidad máxima mayor puede dar como resultado un modelo más complejo.

El **número mínimo de observaciones requeridas en una hoja** es **2**. Esto significa que una hoja del árbol solo se creará si contiene al menos 2 ejemplos. Este hiperparámetro ayuda a evitar que el árbol se ajuste demasiado a ruido en los datos.

El **número mínimo de observaciones requeridas para dividir un nodo interno** es **6**. Esto significa que un nodo se dividirá en dos si contiene al menos 6 ejemplos. Este hiperparámetro controla la división de nodos y puede ayudar a evitar divisiones poco significativas.

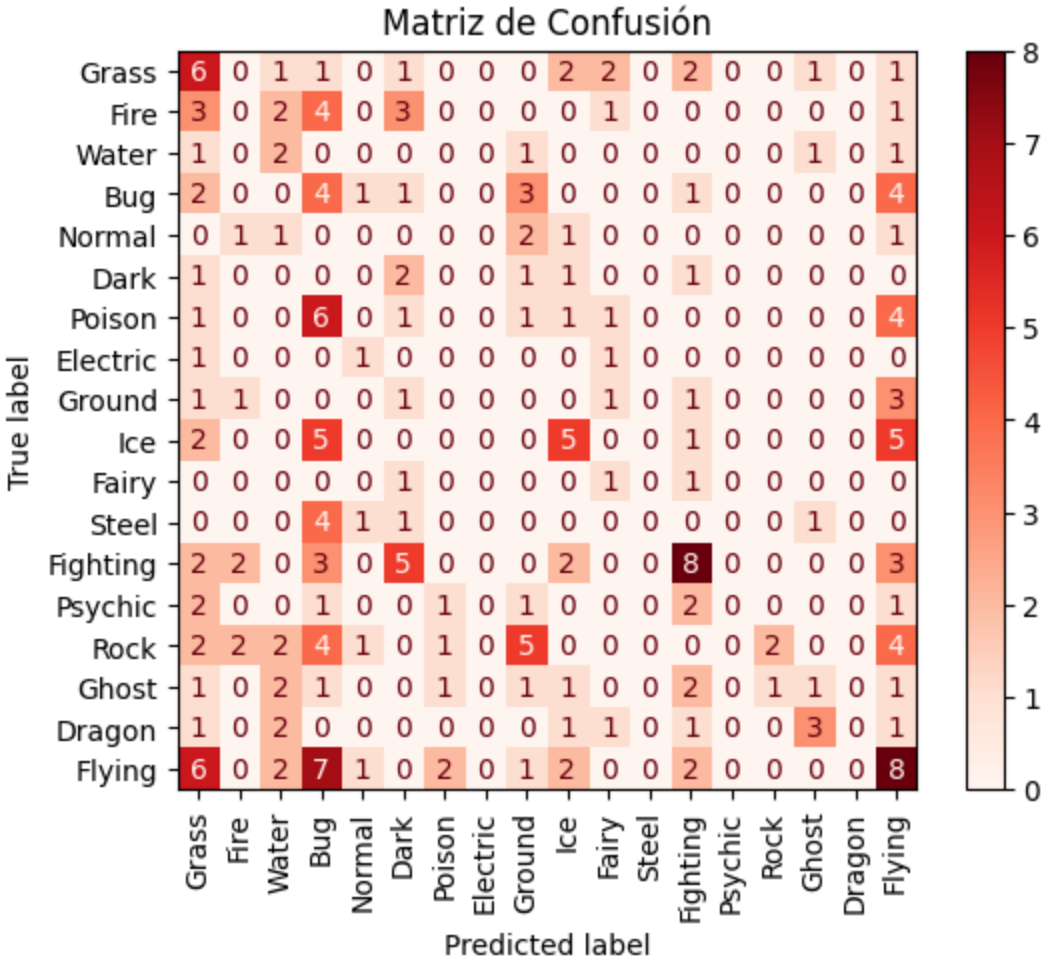
## 2.2.4. Visualización del árbol

En esta sección se proporciona una representación gráfica y comprensible del modelo de árbol de decisión, mostrando las reglas y criterios utilizados para tomar decisiones. Esto facilita la interpretación y comprensión del funcionamiento del modelo en términos de decisiones basadas en características específicas.

Para ver la imagen con más claridad se puede correr el Jupyter Notebook del trabajo y se creará un archivo “arbol_decision_clasificacion.pdf” con el gráfico vectorizado.

## 2.2.5. Matriz de confusión

La matriz de confusión permite examinar el rendimiento del modelo en términos de aciertos y errores en la clasificación de clases, lo que contribuye a una comprensión más profunda de su capacidad predictiva.



**Verdaderos positivos (TP): Elementos en la diagonal principal**
Estos valores representan la cantidad de instancias de cada clase que el modelo ha clasificado correctamente. Cuanto más altos sean estos valores en la diagonal principal, mejor será el rendimiento del modelo.

En la matriz de confusión del árbol de decisión podemos ver una diagonal poco poblada, lo que indica un bajo rendimiento en general. En particular, se destaca una mayor tasa de aciertos en los tipos Planta, Hielo, Lucha y Volador.

**Falsos negativos (FN): Elementos en un fila no diagonales**
Estos valores indican la cantidad de instancias de una clase específica que el modelo ha clasificado incorrectamente como otra clase. Un valor alto de FP puede sugerir que el modelo tiene dificultades para distinguir entre esas dos clases.

Otra vez puede observarse un bajo rendimiento en este punto, por ejemplo, a pesar de haber acertado una gran cantidad de casos para el tipo Volador (8), también clasificó incorrectamente en otras clases a una gran cantidad de instancias (6+0+2+7+1+0+2+0+1+2+0+0+2+0+0+0+0 = 23).

**Falsos positivos (FP): Elementos en una columna no diagonales**
Estos valores indican la cantidad de instancias que el modelo ha clasificado incorrectamente como de una clase pero en realidad resultan ser de otra. Un valor alto de FP señala que el modelo para identificar instancias de esa clase en particular.

En este punto el modelo también tiene falencias ya que, utilizando el mismo ejemplo que antes, si bien el modelo clasificó correctamente 8 instancias de pokemon volador, también clasificó en esta clase una gran cantidad de instancias que en realidad, pertenecían a otra
(1+1+1+4+1+0+4+0+3+5+0+0+3+1+4+1+1 = 30)

## 2.2.6 - Métricas del modelo

En esta sección, se calculan y muestran diversas métricas de evaluación del modelo de clasificación basado en un árbol de decisión. Las métricas calculadas son precisión, exactitud y exhaustividad.

Los resultados fueron los siguientes:

**Precisión (Precision): 0.198**
La precisión en un problema multiclase mide la proporción de predicciones correctas para una clase específica en relación con todas las predicciones realizadas para esa clase. En este caso, la precisión promedio para todas las clases es del 19.8%. Esto significa que, en promedio, solo el 19.8% de las

predicciones para cada clase son correctas. Una precisión baja indica que el modelo tiende a hacer muchas predicciones incorrectas para las clases.

**Exhaustividad (Recall): 0.171**
La exhaustividad (también conocida como recall) en un problema multiclase mide la proporción de ejemplos reales de una clase específica que se han predicho correctamente en relación con el total de ejemplos reales de esa clase. En este caso, la exhaustividad promedio para todas las clases es del 17.1%. Esto significa que, en promedio, solo el 17.1% de los ejemplos reales para cada clase se predicen correctamente. Una exhaustividad baja indica que el modelo tiende a perder muchos ejemplos positivos reales.

**Exactitud (Accuracy): 0.171**
La exactitud en un problema multiclase mide la proporción de todas las predicciones correctas en relación con el total de predicciones realizadas. En este caso, la exactitud promedio para todas las clases es del 17.1%. Esto significa que, en promedio, solo el 17.1% de todas las predicciones son correctas. Una exactitud baja indica que el modelo no se desempeña bien en la clasificación de las clases.

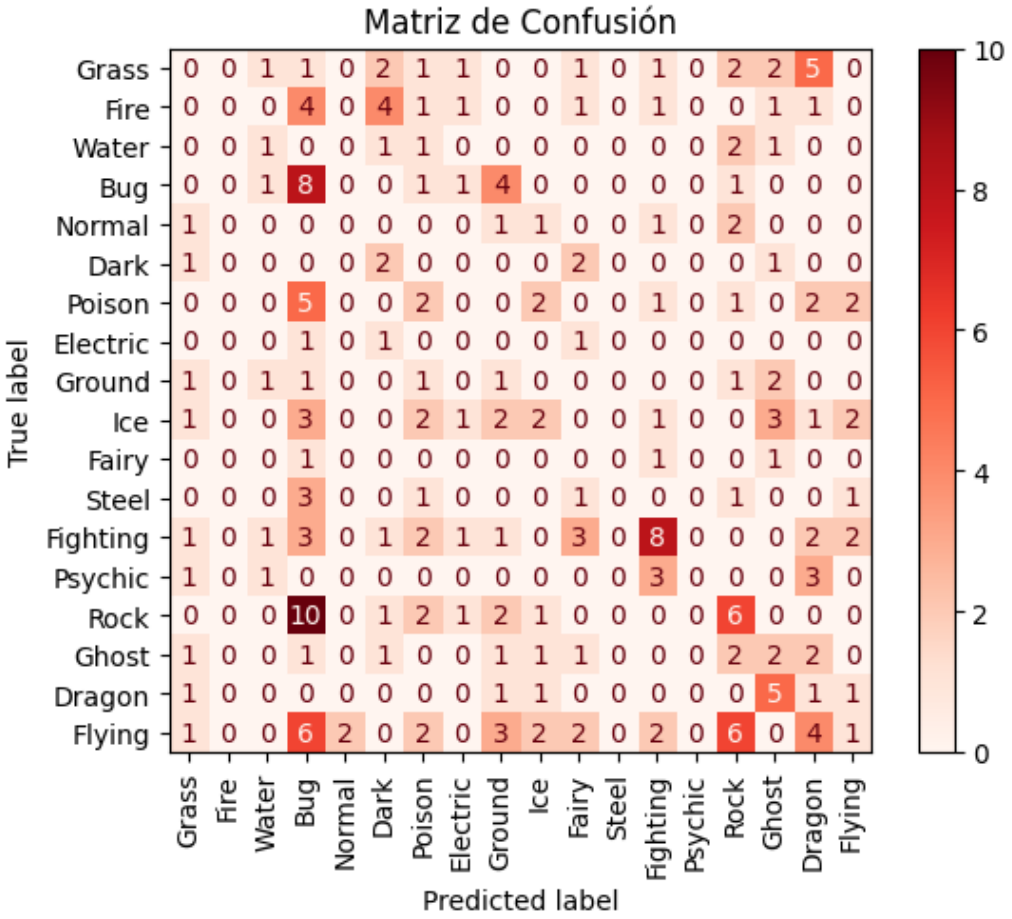
## 2.3. Bayes Ingenuo (Naive Bayes)

### 2.3.1. Procedimiento

Como en los demás modelos se separaron los datos en un conjunto de entrenamiento y otro de prueba. Se escogió el clasificador Multinomial de Naive Bayes por la naturaleza de los datos, a saber, números enteros no negativos. Se creó una instancia de la clase MultinomialNB de Sckit-learn y se entrenó el modelo.

### 2.3.2. Matriz de confusión

En la matriz de confusión multiclase presentada a continuación, es evidente que una fracción minoritaria de los datos se sitúa en la diagonal de la matriz, la cual representa predicciones acertadas. Esto sugiere un desempeño subóptimo en la evaluación. No obstante, para obtener una comprensión más precisa y detallada de su rendimiento, se llevará a cabo el cálculo de métricas específicas en la siguiente sección.




## 2.3.3. Métricas de Bayes Ingenuo Multinomial

Se calcularon las mismas métricas que para el modelo de clasificación con árbol de decisión. Los resultados fueron los siguientes:

**Precision (Precisión)**
El valor de precisión es 0.141, lo que significa que solo alrededor del 14.1% de las predicciones positivas realizadas por el modelo fueron correctas. En otras palabras, el modelo tiene una alta tasa de falsos positivos.

**Accuracy (Exactitud)**
La exactitud es de 0.149, lo que indica que el modelo acierta aproximadamente el 14.9% de las predicciones en general. Sin embargo, una exactitud baja podría deberse a un desequilibrio en las clases objetivo o a un rendimiento deficiente del modelo.

**Recall (Exhaustividad)**
El valor de recall es 0.149, lo que significa que solo se capturan aproximadamente el 14.9% de todos los casos verdaderos positivos. El recall es bajo, lo que indica que el modelo no es efectivo en la identificación de casos positivos.

## 2.4. Clasificación con K-NN (k-Nearest Neighbors)

### 2.4.1. Estandarización

En esta sección, se procederá a la estandarización de los datos en preparación para la aplicación del algoritmo K-NN. La estandarización es un paso esencial para asegurar que todas las características tengan un rango y escala comparables, lo que facilita el cálculo de las distancias entre puntos, un aspecto crítico en el funcionamiento efectivo del algoritmo K-NN.

Se ha elegido la técnica de escalamiento Min-Max. Esta elección se basa en varios factores clave que incluyen la falta de una distribución normal en los datos, lo cual descarta la aplicación de la transformación Z-Score. Además, la eliminación de valores atípicos previamente realizada descarta el uso de un enfoque robusto, y la inexistencia de una relación logarítmica en los datos hace inapropiado el empleo de la técnica homónima.

### 2.4.2. Procedimiento

Se separaron los datos en conjuntos de entrenamiento y de prueba, se creó una instancia de la clase KNeighborsClassifier de Scikit-Learn y se optimizaron los hiperparámetros con  GridSearchCV de la misma librería.

### 2.4.3. Hiperparámetros óptimos

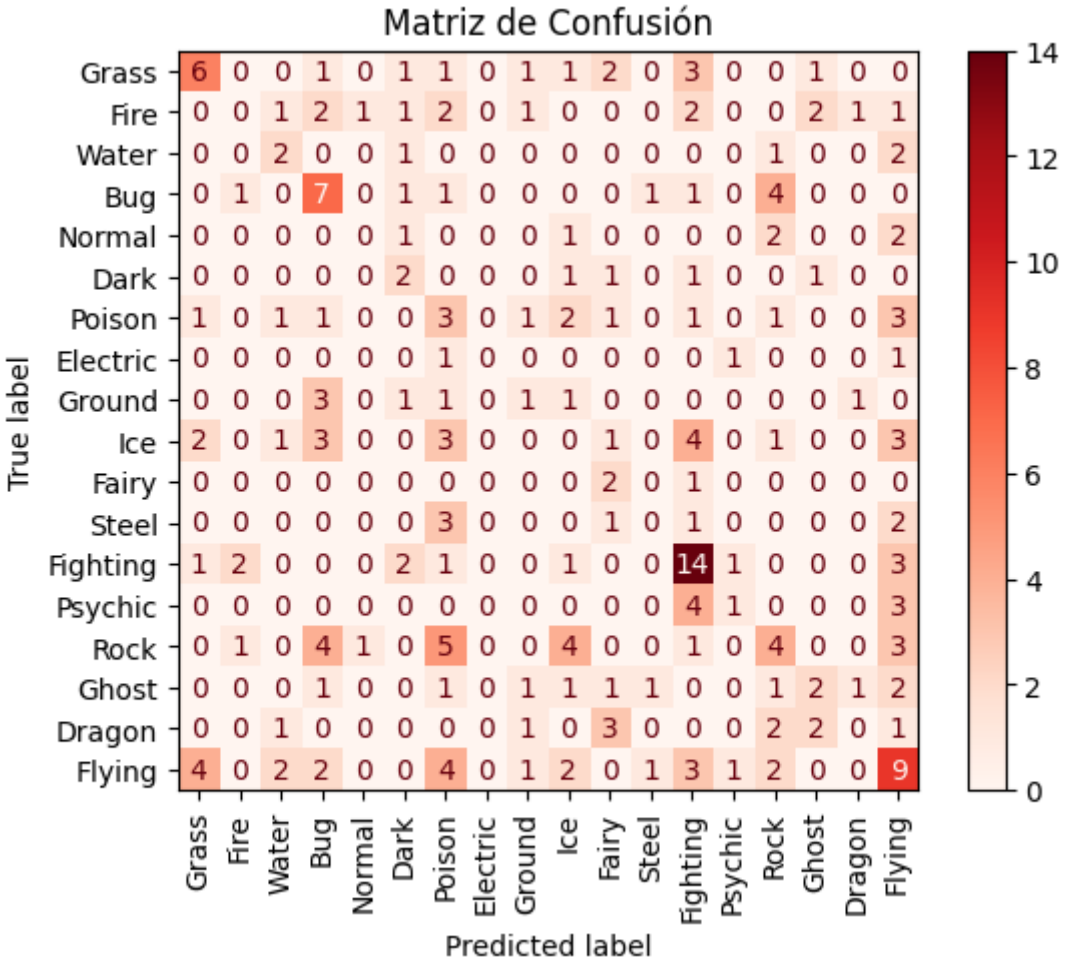
**n_neighbors: 13**
Este hiperparámetro representa el número de vecinos más cercanos que se tomarán en cuenta al realizar una predicción para una nueva instancia. En este caso, el valor óptimo es 13, lo que significa que se considerarán los 13 vecinos más cercanos para tomar decisiones de clasificación.

**p: 1**
Este hiperparámetro corresponde a la métrica de distancia utilizada en el algoritmo KNN. Cuando p es igual a 1 (como en este caso), se está utilizando la distancia de Manhattan (también conocida como L1 norm), que mide la distancia como la suma de las diferencias absolutas de las coordenadas.

**weights: 'distance'**
Este hiperparámetro determina cómo se ponderan los vecinos al realizar una predicción. En este caso, se ha elegido 'distance', lo que significa que los vecinos más cercanos tienen un peso más alto en la predicción en función de la inversa de su distancia. Esto da más importancia a los vecinos más cercanos y puede mejorar la precisión del modelo.

## 2.4.4. Matriz de confusión

En el modelo K-Nearest Neighbors (KNN), se destaca una mayor concentración de instancias en la diagonal principal de la matriz de confusión, con una particular prominencia en el tipo "Lucha". No obstante, se observa que persiste un número significativo de instancias mal clasificadas en otras categorías.




## 2.4.5. Métricas de K-NN

**Precision (Precisión)**

En este caso, la precisión es 0.201. Esto significa que de todas las predicciones que el modelo hizo para una clase específica, el 20.1% de esas predicciones fueron correctas. En otras palabras, el modelo tiene una tasa de falsos positivos relativamente alta.

**Accuracy (Exactitud)**

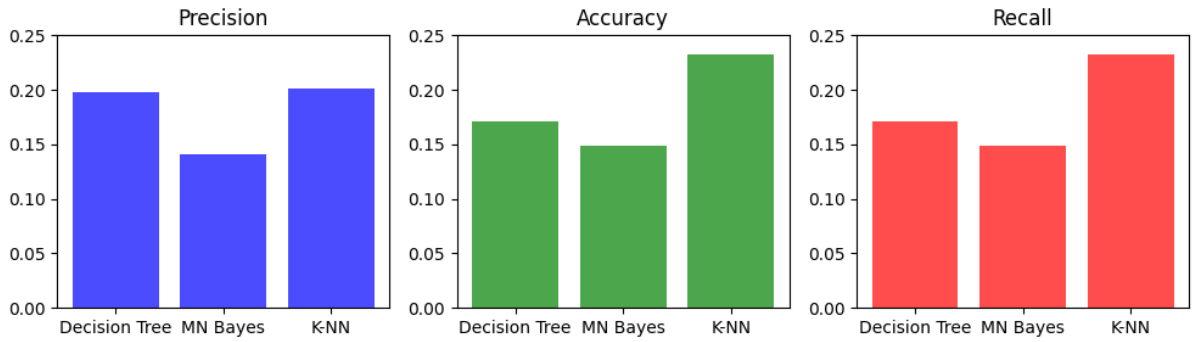
La exactitud es 0.232, lo que indica que el modelo clasifica correctamente el 23.2% de las muestras en el conjunto de prueba. Esto es bastante bajo y sugiere que el modelo no es muy preciso en general.

**Recall (Ehaustividad)**
La exhaustivdad es 0.232, lo que significa que el modelo identifica correctamente el 23.2% de todas las muestras de una clase específica en el conjunto de prueba. En otras palabras, el modelo no es muy bueno para identificar positivos verdaderos.

## 2.5. Resultados

A continuación se muestran una tabla comparativa de las métricas para cada modelo de clasificación y un gráfico de barras.

| | Modelo | Precision | Accuracy | Recall |
|---|---|---|---|---|
| 0 | Decision Tree | 0.197629 | 0.171053 | 0.171053 |
| 1 | MN Bayes | 0.141017 | 0.149123 | 0.149123 |
| 2 | K-NN | 0.201012 | 0.232456 | 0.232456 |




Puede verse como la clasificación por K-NN da un mejor resultado para la clasificación en todas las métricas seguido por el árbol de decisión y en último lugar el Multinomial Naive Bayes.

## 2.6. Conclusión

En el transcurso de este trabajo, se llevaron a cabo implementaciones y comparaciones de tres modelos de clasificación: Árboles de Decisión, Naive Bayes y K-Vecinos Más Cercanos (K-NN), con el propósito de predecir el tipo de un Pokémon en función de sus estadísticas base. El enfoque metodológico abarcó una exhaustiva exploración de los datos, incluyendo una descripción detallada de las columnas, el análisis de medidas de localización y centralidad de los valores, la evaluación de la distribución de los valores de las columnas y el examen de la correlación entre pares de variables.

A continuación, se procedió con la implementación de los modelos mencionados, dividiendo los datos en conjuntos de entrenamiento y prueba. Para mejorar su desempeño, se optimizaron los hiperparámetros de los modelos Árbol de Decisión y K-NN. Posteriormente, se llevaron a cabo análisis de las matrices de confusión de los mejores modelos y se obtuvieron métricas fundamentales, incluyendo Precision (Precisión), Accuracy (Exactitud) y Recall (Exhaustividad).

Los resultados obtenidos revelaron un desempeño generalmente insatisfactorio en lo que respecta a la capacidad de determinar el tipo de Pokémon. El modelo de K-Vecinos Más Cercanos (K-NN) se posicionó como el mejor predictor, seguido por el Árbol de Decisión, y en última instancia, el modelo Naive Bayes.

Se optó por implementar el clasificador Naive Bayes en su variante Multinomial, diseñada para manejar datos enteros no negativos, una característica acorde con la naturaleza de nuestro conjunto de datos. A pesar de esta elección, se exploró también la opción del modelo Gaussiano, cuyo rendimiento resultó incluso menos satisfactorio. Es relevante destacar que se intentó optimizar el hiperparámetro del suavizado Laplaciano (alfa) en el modelo Multinomial Naive Bayes con el propósito de evitar la probabilidad condicional igual a cero. Sin embargo, este esfuerzo resultó infructuoso, ya que el valor óptimo para alfa se mantuvo en 0 de manera consistente.

Un aspecto adicional que podría explicar el bajo rendimiento del modelo Naive Bayes Multinomial radica en la suposición de independencia condicional entre predictores, una característica fundamental del modelo Bayesiano Ingenuo. En este contexto, el análisis exploratorio de datos evidenció la existencia de correlación significativa entre algunos pares de características, lo cual podría haber incidido en el rendimiento del modelo.

Finalmente, se concluye que las estadísticas base de los Pokémon podrían no ser un indicador idóneo de su tipo. A pesar de esto, entre los modelos evaluados, K-Vecinos Más Cercanos (K-NN) destaca como la elección preferente para abordar esta tarea, aunque se reconoce que aún existen desafíos en la mejora del rendimiento, los cuales pueden requerir un enfoque más sofisticado o la consideración de otras fuentes de información en investigaciones futuras.