

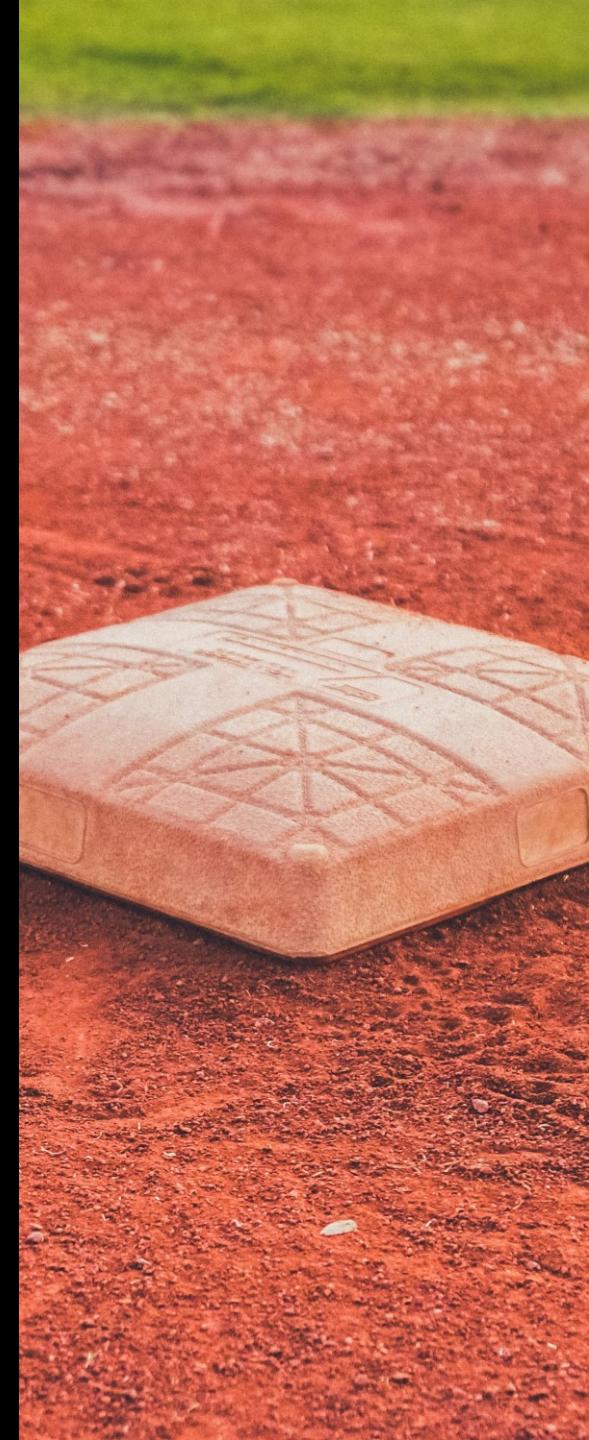
Proyecto Big Data

**MLB Dataset:
Modelo de Clasificación**

Samuel Monge Alvarado
7 de julio del 2022

Planteamiento

- Modelo para predecir los equipos de *Major League Baseball* que clasificarán a Posttemporada.
- Problema de clasificación binaria.
- Datasets: métricas individuales y grupales de los equipos de los últimos 15 años (2006 al 2021).
- Datasets provenientes del repositorio:
[MLB Stats, Scores, History, & Records | Baseball-Reference.com](https://www.baseball-reference.com/).



Datasets extraídos

- **Dataset 1:** Estadísticas de bateo por equipo (19 atributos útiles).
- **Dataset 2:** Estadísticas de bateo por jugador (10 atributos útiles).
- **Dataset 3:** Estadísticas de pitcheo por equipo (18 atributos útiles).
- **Dataset 4:** Estadísticas de pitcheo por jugador (9 atributos útiles).
- **Dataset 5:** Estadísticas de fieldeo por equipo (2 atributos útiles).
- **Dataset 6:** Estadísticas misceláneas por equipo (2 atributos útiles).
- **Dataset 7:** Etiqueta de clasificación a posttemporada por equipo + abreviatura.

Funciones creadas

`df_year = build_df_year(year):`

- *read_file*: directamente desde la jerarquía de folders y archivos.
- *convert_labels*: encode de etiquetas de Posttemporada.
- *remove_rows*: filtro para remover filas de jugadores “ruido”.
- *count_players*: contador de jugadores y agrupador por equipo según atributo y umbral escogidos.
- *joins*: múltiples joins para unión de los subsets (fullouter – inner).

`df_final = union_df(list_df_years):`

Aplicación de 15 veces de la función *build_df_year*.

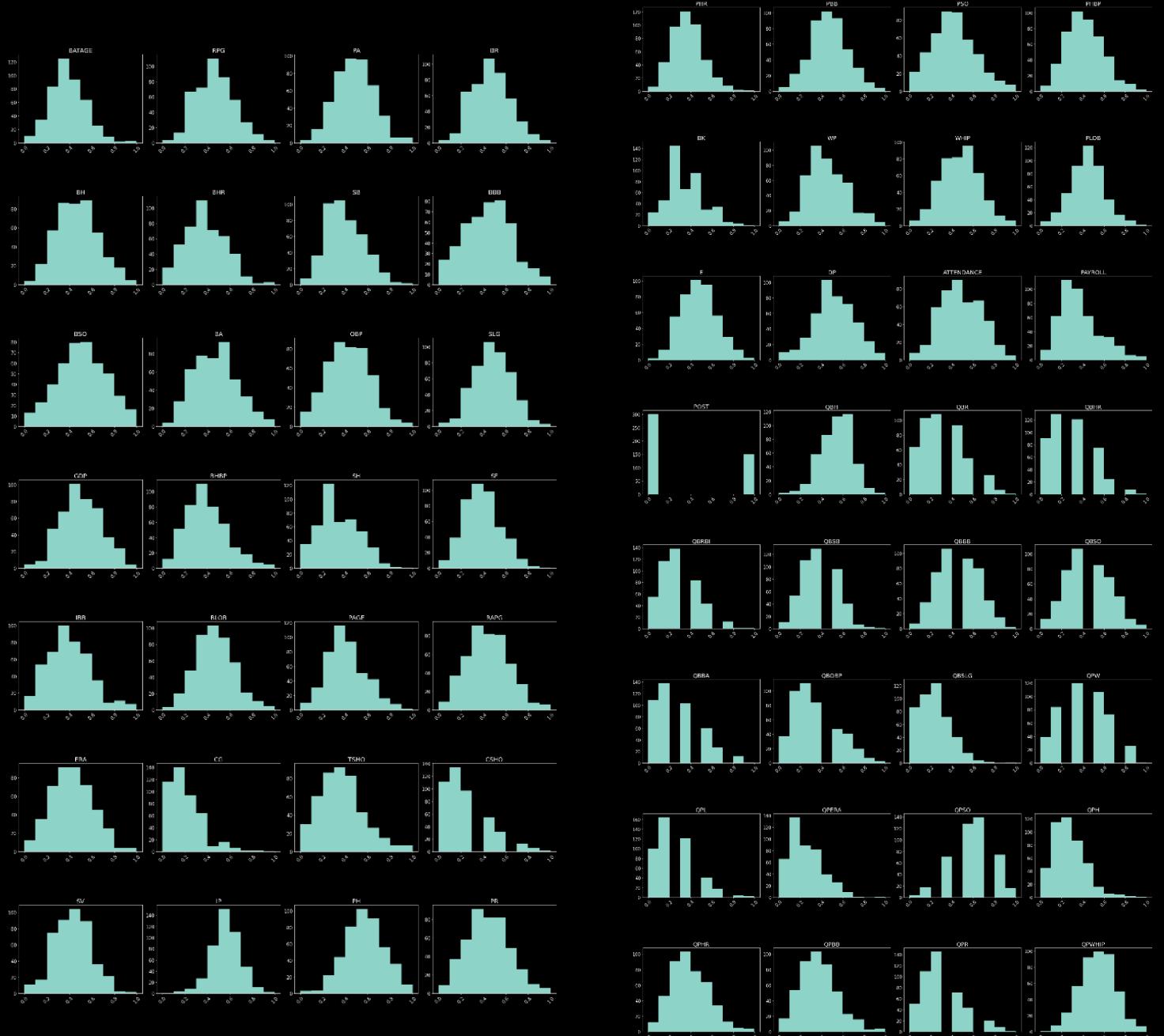
Unión de los dataframes mediante *union*.

Ejecución del Main

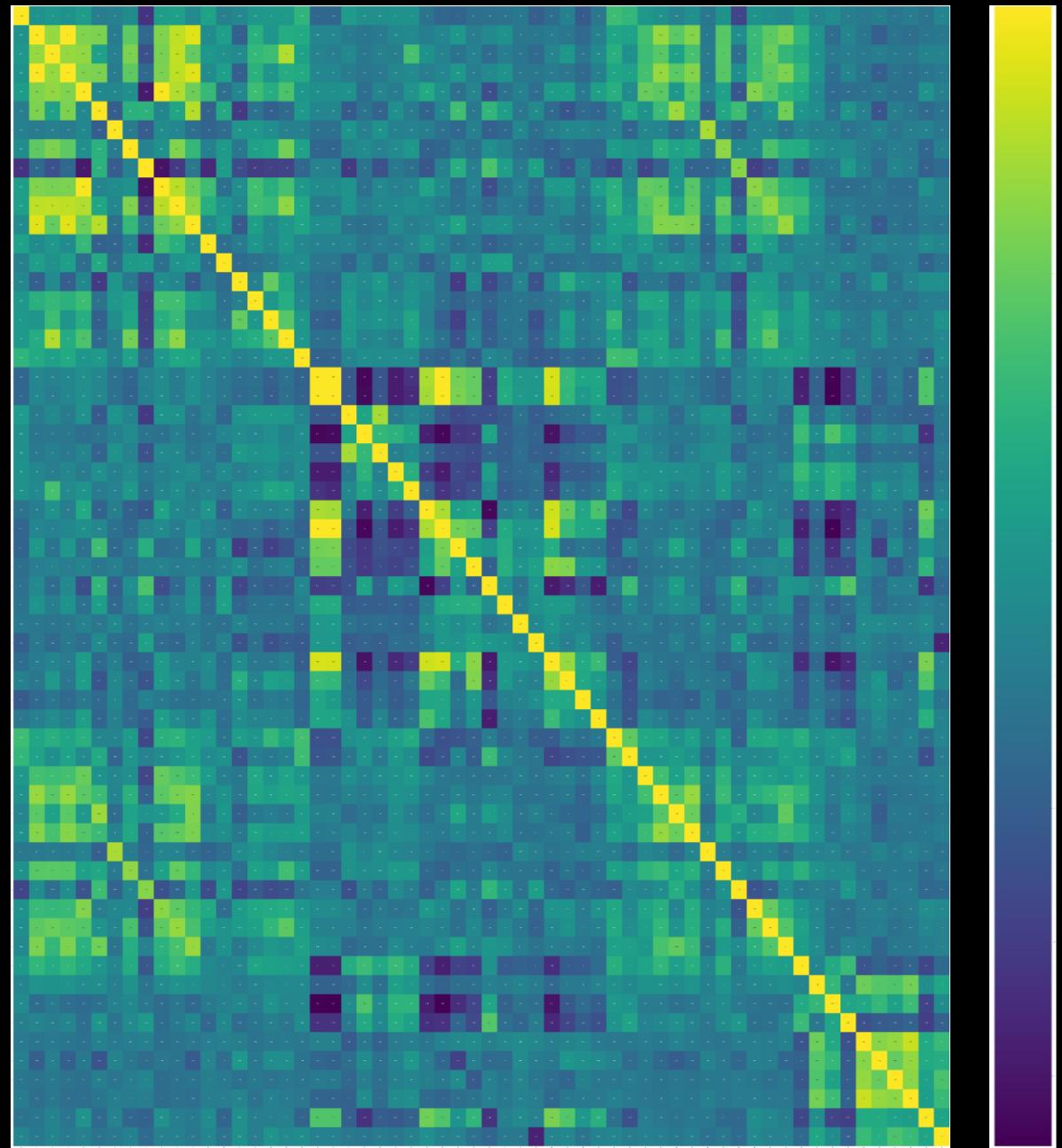
- **Main_1:**
df_final1: contiene DFs del 2021 al 2016.
- **Main_2:**
df_final2: contiene DFs del 2015 al 2011.
- **Main_3:**
df_final3: contiene DFs del 2010 al 2006.

Culmina con escritura en DB de Postgresql y genera archivo csv.

Histogramas de los 60 atributos



Matriz de Correlaciones de los 60 atributos





Desarrollo de modelos de ML

División de training y test:
70%/30%.

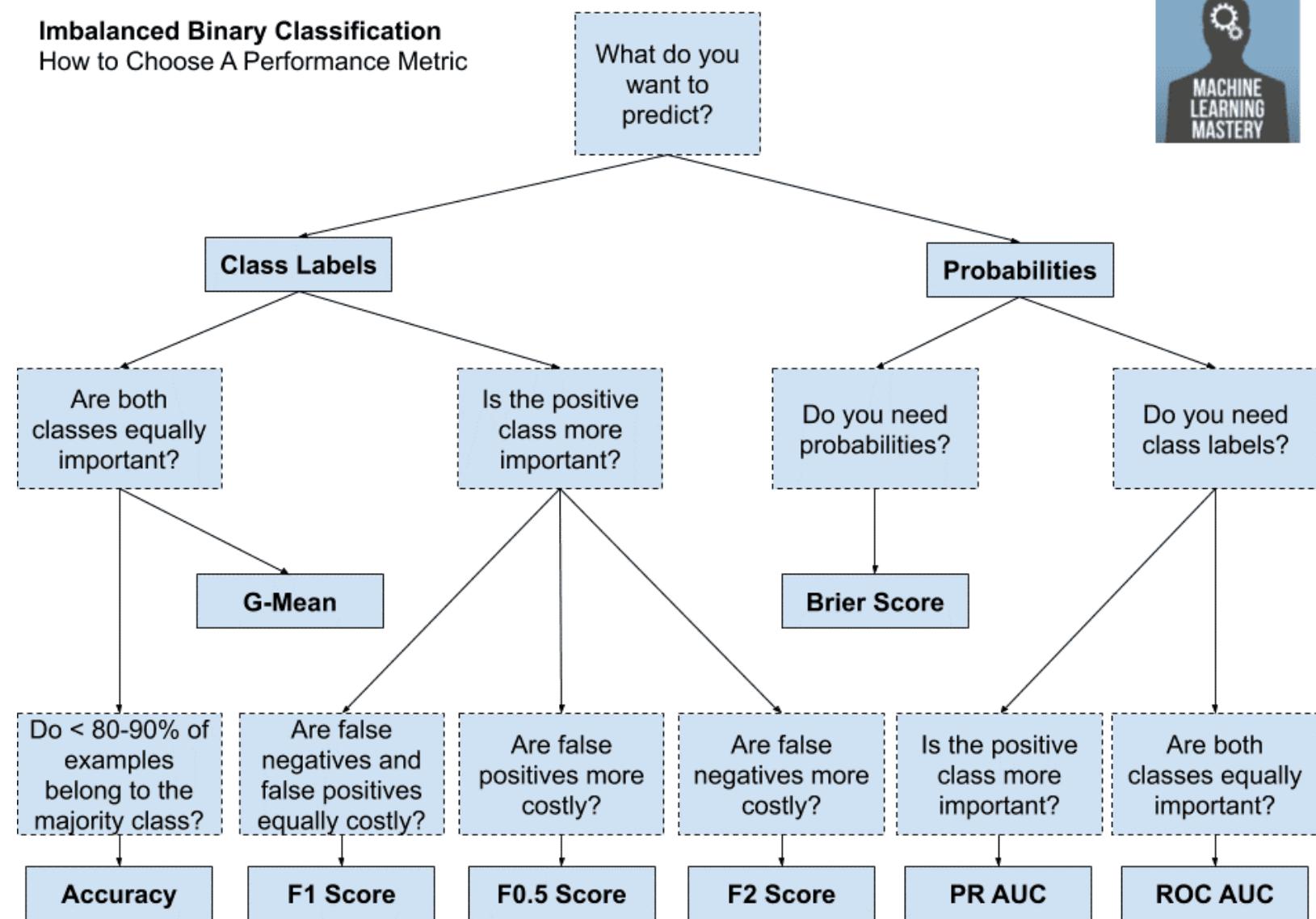
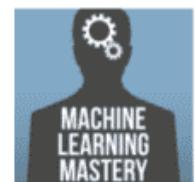
Métrica de clasificación:

F0.5 Score

Validación cruzada:
4 pliegues

label	count
1.0	41
0.0	92

Imbalanced Binary Classification How to Choose A Performance Metric



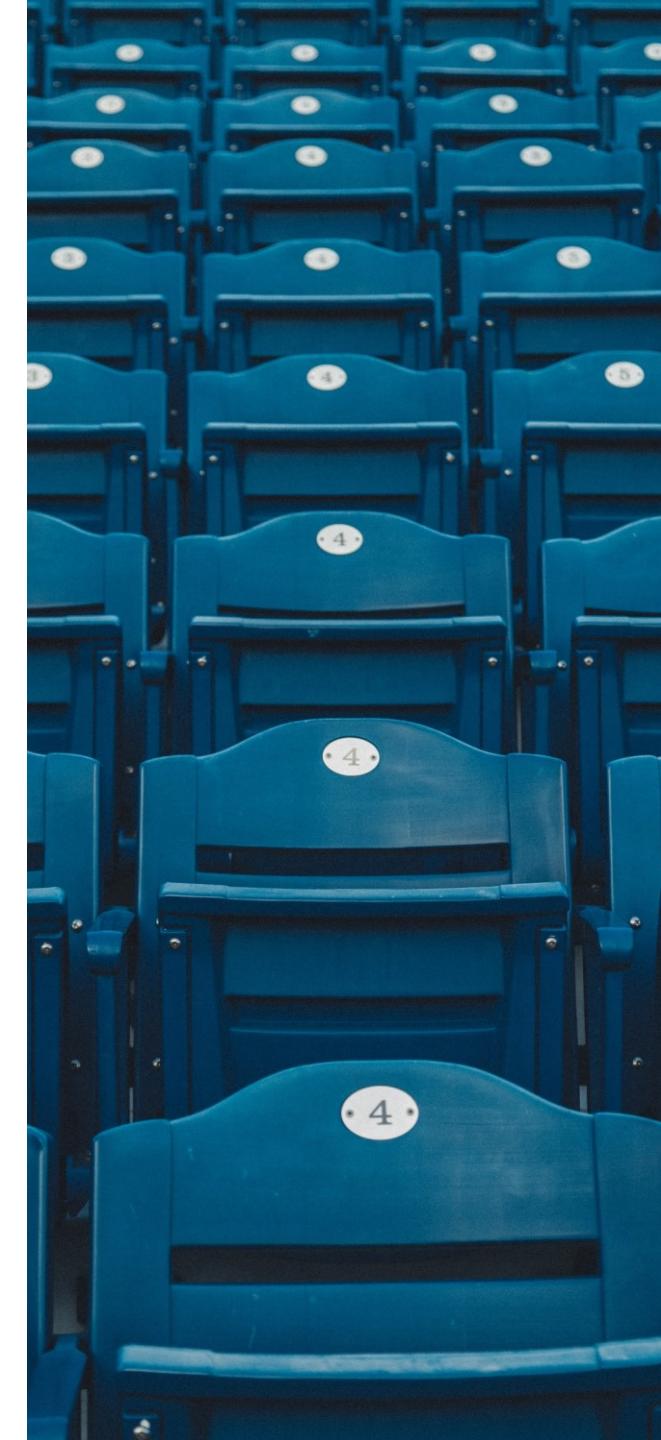
Modelo 1

Multilayer Perceptron Classifier

- 2 capas intermedias de 100 neuronas c/u.
- maxIter: 300 iteraciones.
- blockSize: 32 inputs.

Training F0.5-Score: 100%

Test F0.5-Score: 83.8%



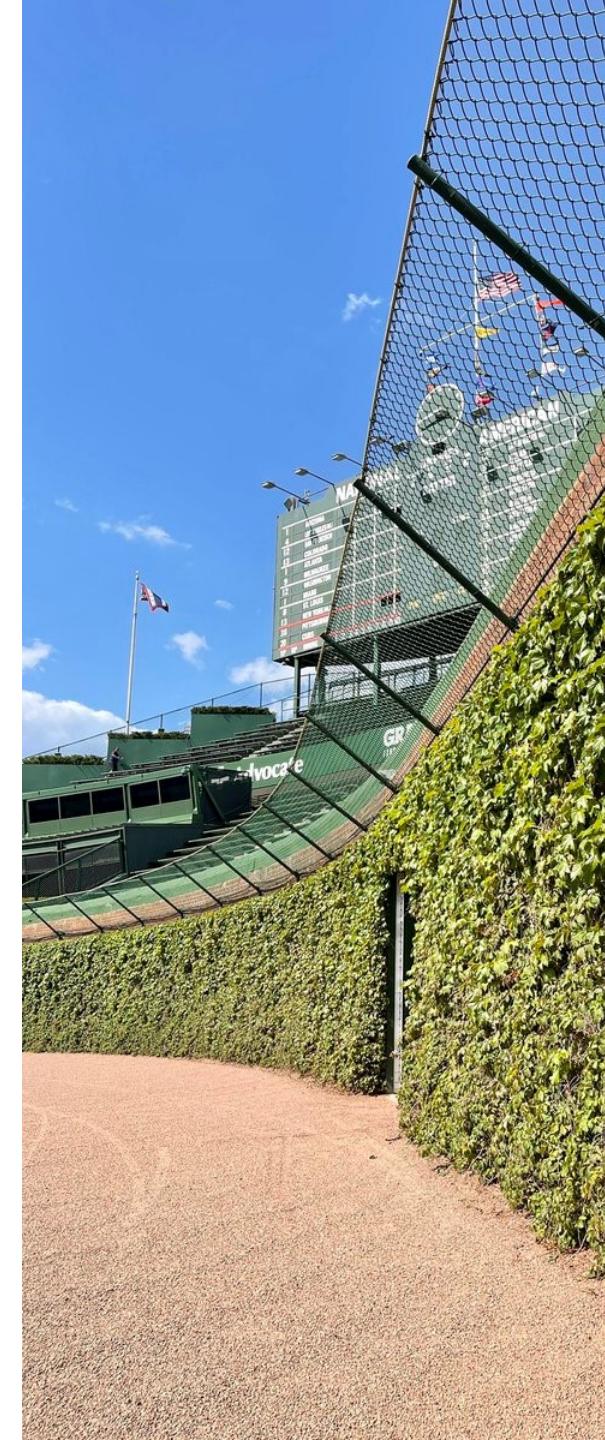
Modelo 2

Random Forest Classifier

- numTrees: 250 árboles.
- maxDepth: 5 niveles por árbol.
- featureSubsetStrategy: 7 features por árbol.

Training F0.5-Score: 98.5%

Test F0.5-Score: 88.1%



Modelo 3

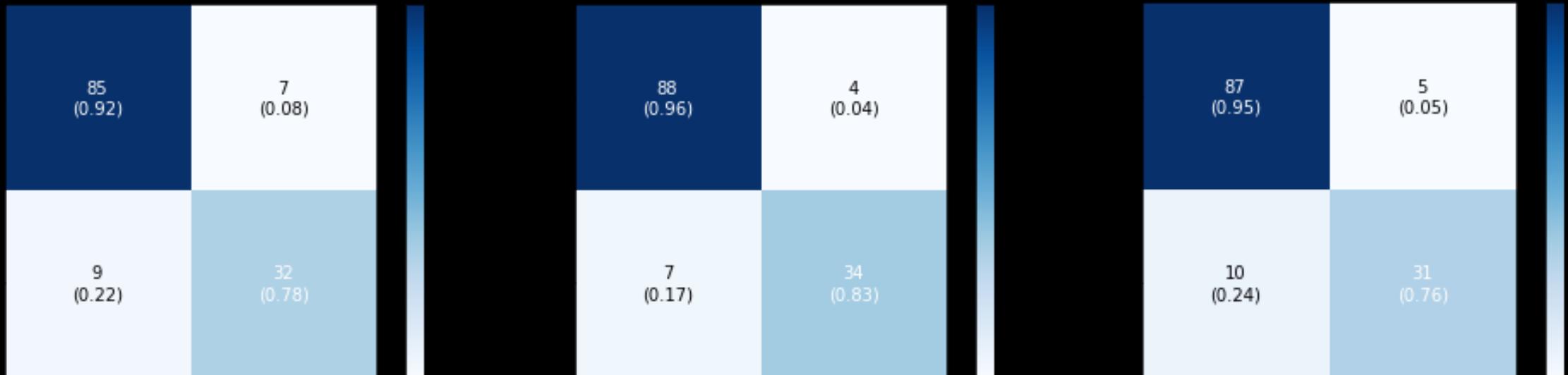
Gradient Boosted Tree Classifier

- maxIter: 150 iteraciones de boosting.
- maxDepth: 5 niveles por árbol.
- maxBins: 10 bins para split de árboles.

Training F0.5-Score: 100%

Test F0.5-Score: 81.2%





Multilayer Perceptron

F0.5 Score: 83.8%

Diferencial train-test: 16.2%

Random Forest

F0.5 Score: 88.1%

Diferencial train-test: 10.4%

Gradient Boosted Tree

F0.5 Score: 81.2%

Diferencial train-test: 18.8%

Comparativa de Modelos

Próximos Pasos

- Agregar más atributos individualizados por jugador.
- Aplicar una técnica de selección de Features.
- Extraer en detalle features de mayor conveniencia por modelo.
- Probar con otros métodos de ensamble para clasificación.

Proyecto Big Data

