

# Pokémon with stats Cas Kaggle

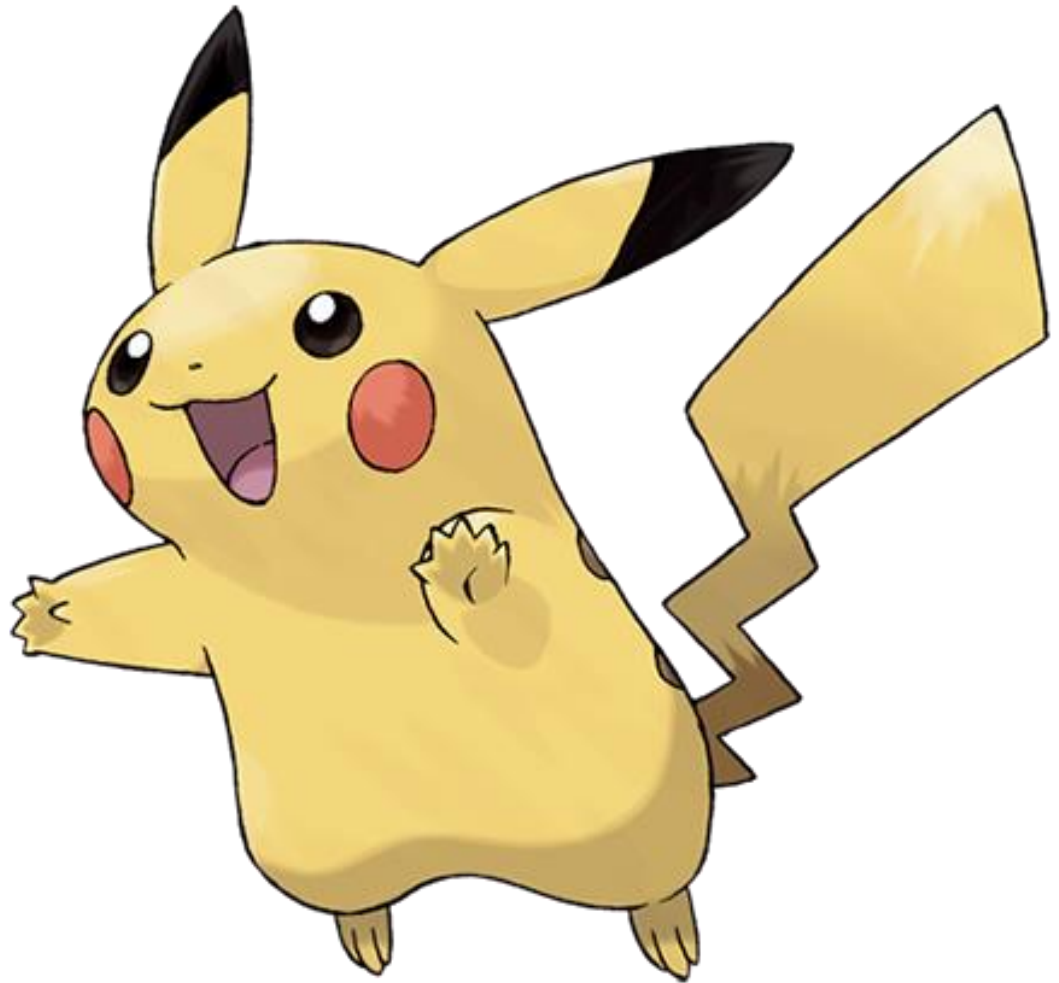
---

Hernán Capilla Urbano

1462773

GitHub: <https://github.com/hcapilla/casKaggle>

Dataset: <https://www.kaggle.com/abcsds/pokemon>



# Introducció

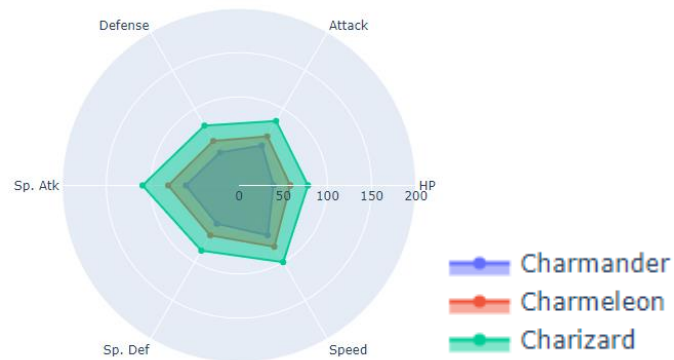
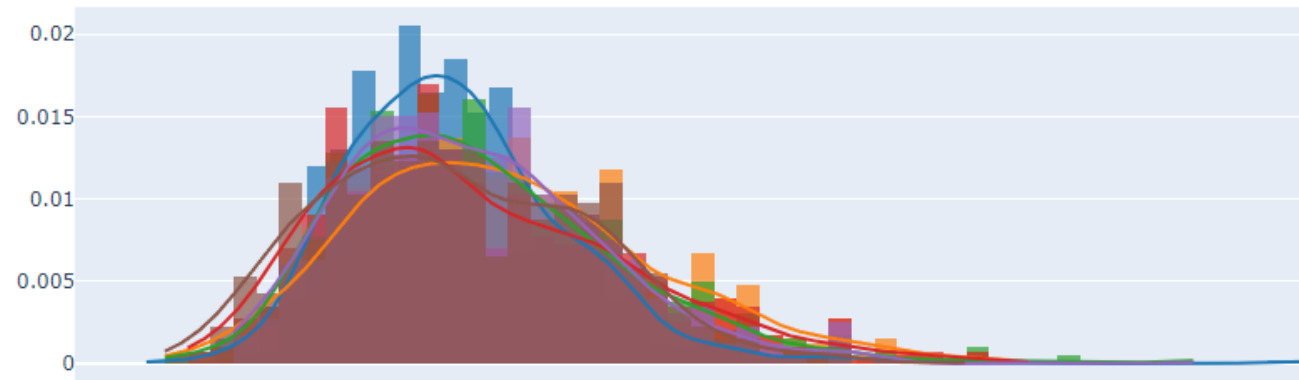
#		Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False



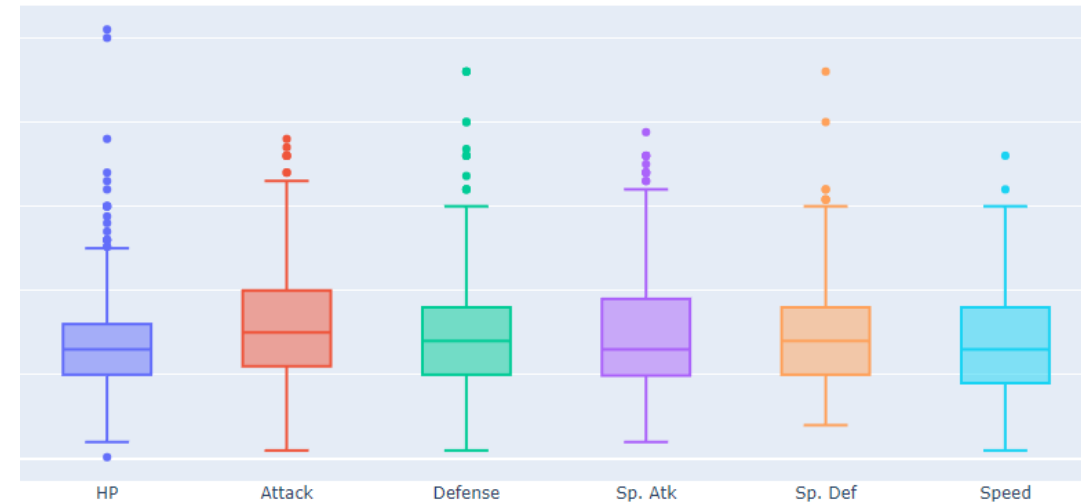
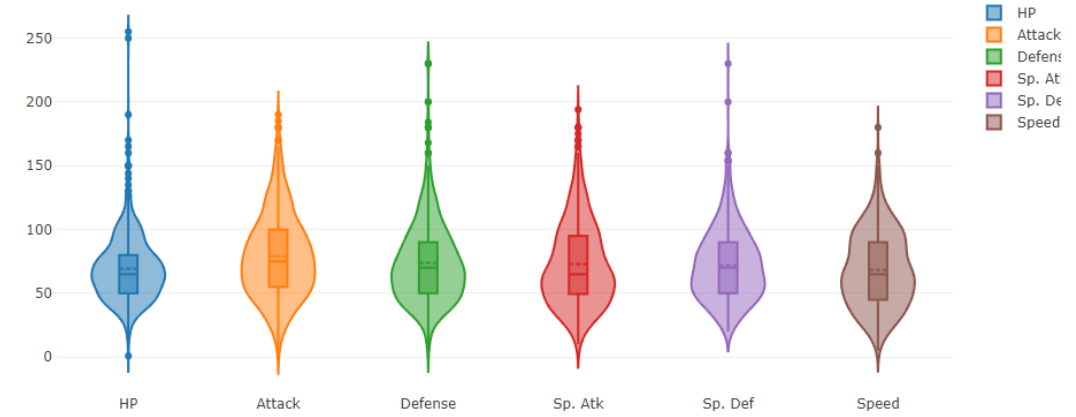
#		Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	Blank	309	39	52	43	60	50	65	1	False

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 13 columns):
#   Column      Non-Null Count  Dtype
---  -
0   #           800 non-null   int64
1   Name        800 non-null   object
2   Type 1      800 non-null   object
3   Type 2      414 non-null   object
4   Total       800 non-null   int64
5   HP          800 non-null   int64
6   Attack      800 non-null   int64
7   Defense     800 non-null   int64
8   Sp. Atk     800 non-null   int64
9   Sp. Def     800 non-null   int64
10  Speed       800 non-null   int64
11  Generation  800 non-null   int64
12  Legendary   800 non-null   bool
dtypes: bool(1), int64(9), object(3)
memory usage: 75.9+ KB
```

# Anàlisi de dades

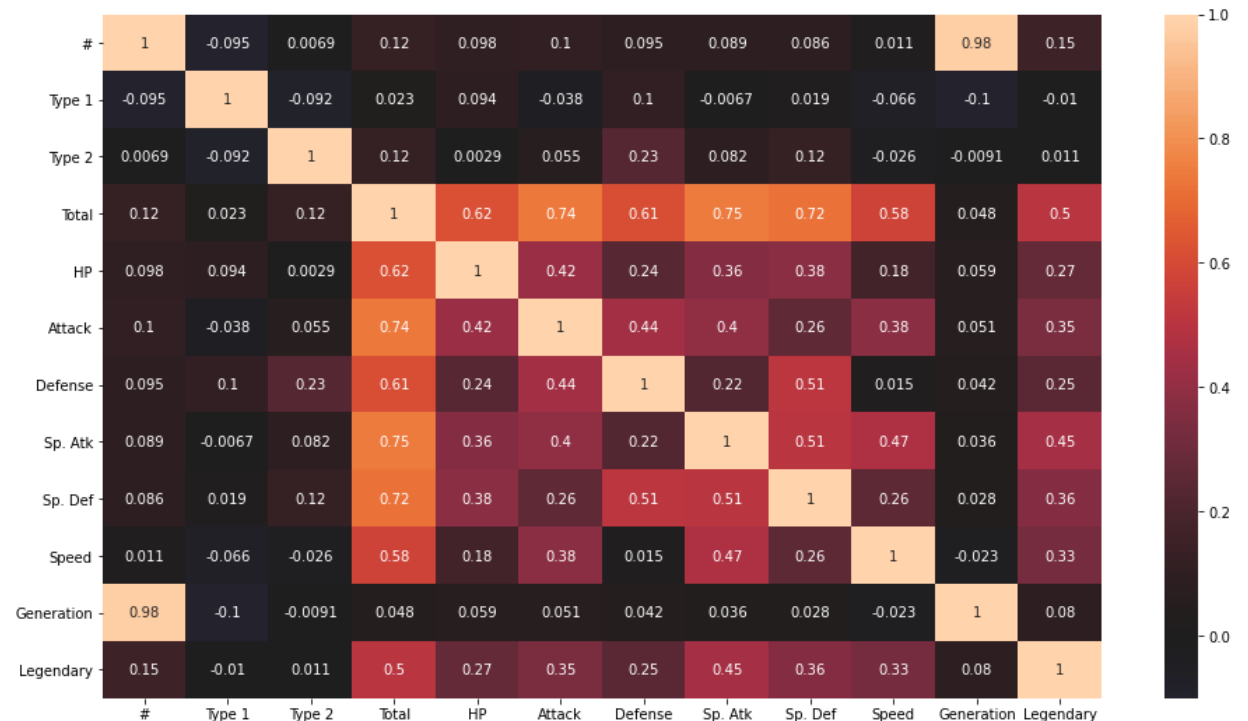
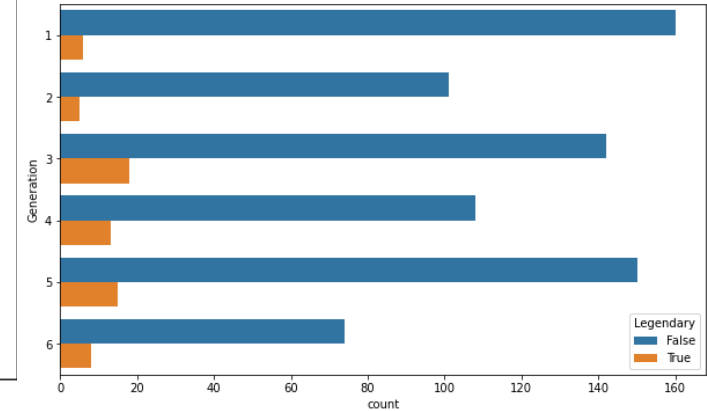
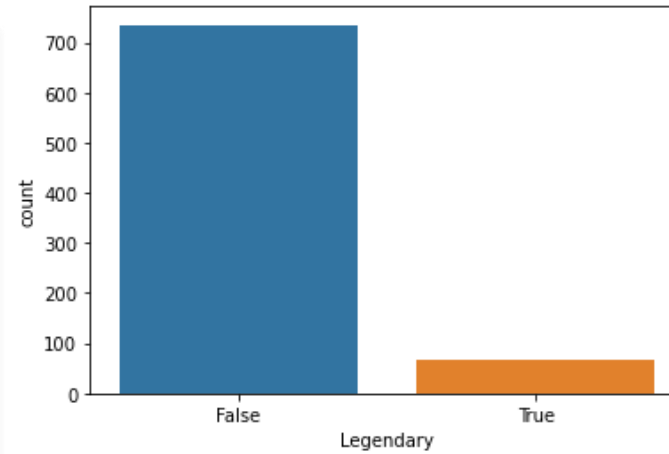


Gràfic de violín de cada estadística básica



# Mètodes d'aprenentatge

- Regresión logística
- Regresión Logística con procesamiento
- SVM
- Random Forest
- Gradient Boosting

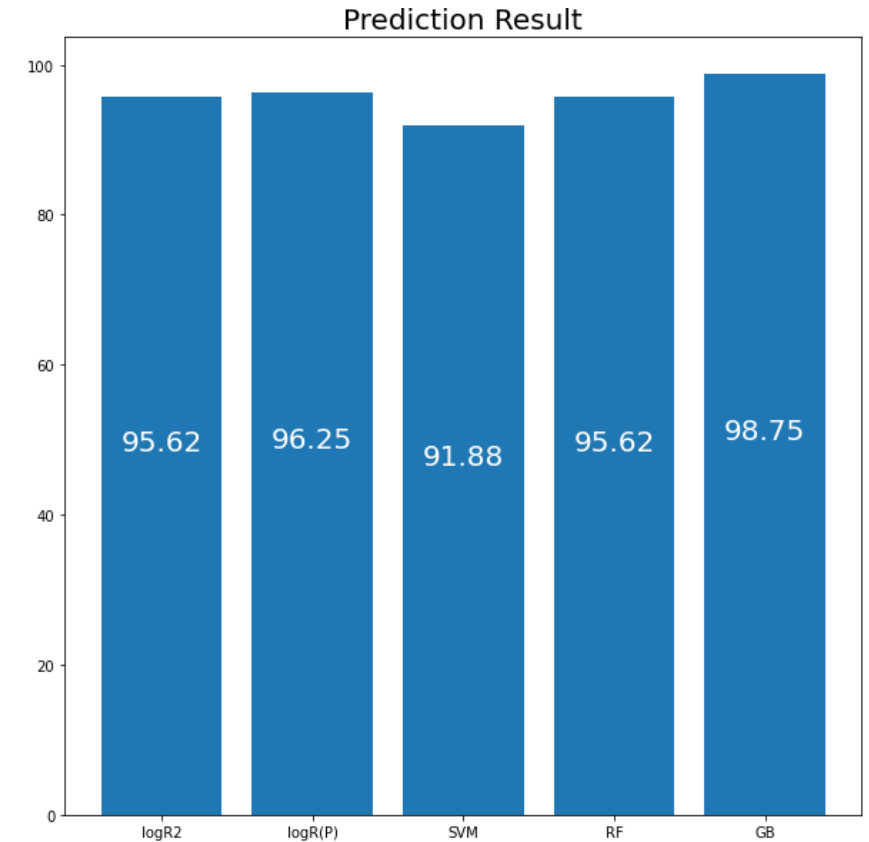


# Busca de millors paràmetres i resultats

- RL → max\_iter = 100 → 91.25
- RL → max\_iter = 200 → 95.625

```
col_trans = make_column_transformer(  
    (OneHotEncoder(), ['Type 1', 'Type 2', 'Generation']),  
    (StandardScaler(), ['Total', 'HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def', 'Speed']), #Escala los datos numéricos  
    remainder = 'passthrough') #Para las demás columnas no haremos nada
```

- RL + trans → 96.25



# Millor resultat i conclusió

- GB → learning rate = 1.1 → 86.875 %
  - GB → n\_estimators = 200 → 96.875 %
  - GB superior:
    - learning\_rate = 0.9
    - n\_estimators = 100
    - validation\_fraction = 0.2
    - n\_iter\_no\_change = 100
    - random\_state = 10 (no paràmetre)
- 98.75%