SCIENCES SORBONNE UNIVERSITÉ

SPLEX PROJECT PRESENTATION

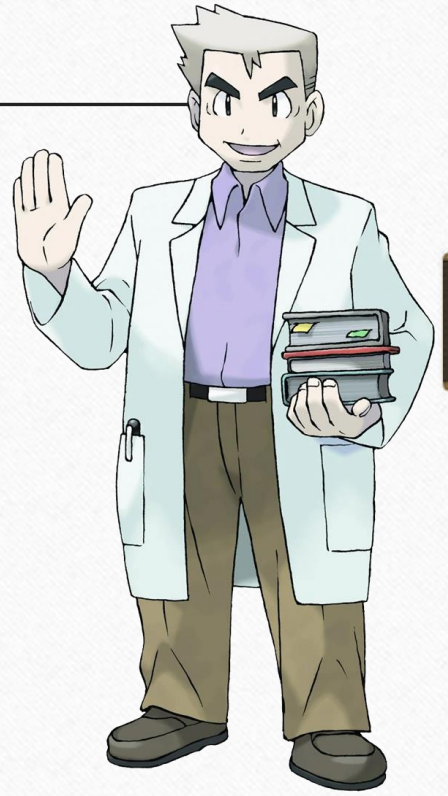
Data exploration
Propose and Develop an optimal method
A Pokémon analysis

LEGOFFIC Liam, **ZHONG** Yann

28/01/2022

TABLE OF CONTENTS

TABLE OF CONTENTS



INTRODUCTION

The data of Pokémon

- Source: kaggle.com/alopez247/pokemon and kaggle.com/rounakbanik/pokemon
- Motivation ➔ 8 "generations" of creatures, all with unique features

Pikachu

Type: Electric
Abilities: Lightning Rod, Static
Tier: NFE

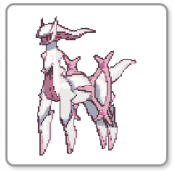
HP:	35
Attack:	55
Defense:	40
Sp. Atk:	50
Sp. Def:	50
Speed:	90

Nihilego

Type: Rock, Poison
Abilities: Beast Boost
Tier: UU

HP:	109
Attack:	53
Defense:	47
Sp. Atk:	127
Sp. Def:	131
Speed:	103

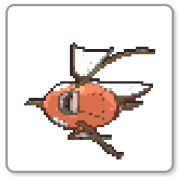
Arceus-Fairy



Type: Fairy
Abilities: Multitype
Tier: Uber

HP:	120
Attack:	120
Defense:	120
Sp. Atk:	120
Sp. Def:	120
Speed:	120

Magikarp



Type: Water
Abilities: Rattled, Swift Swim
Tier: LC

HP:	20
Attack:	10
Defense:	55
Sp. Atk:	15
Sp. Def:	20
Speed:	80

- Attack
- Defense
- Sp. Atk
- Sp. Def
- Speed
- Total
- Exp. Growth
- Typing
- …

INTRODUCTION

The data of Pokémon

❖ Source: kaggle.com/alopez247/pokemon and kaggle.com/rounakbanik/pokemon

❖ Motivation ➔ 8 "generations" of creatures, all with unique features

Pikachu

Type: Electric
Abilities: Lightning Rod, Static
Tier: NFE

HP:	35
Attack:	55
Defense:	40
Sp. Atk:	50
Sp. Def:	50
Speed:	90

Nihilego

Type: Rock, Poison
Abilities: Beast Boost
Tier: UU

HP:	109
Attack:	53
Defense:	47
Sp. Atk:	127
Sp. Def:	131
Speed:	103

Arceus-Fairy

Type: Fairy
Abilities: Multitype
Tier: Uber

HP:	120
Attack:	120
Defense:	120
Sp. Atk:	120
Sp. Def:	120
Speed:	120

Magikarp

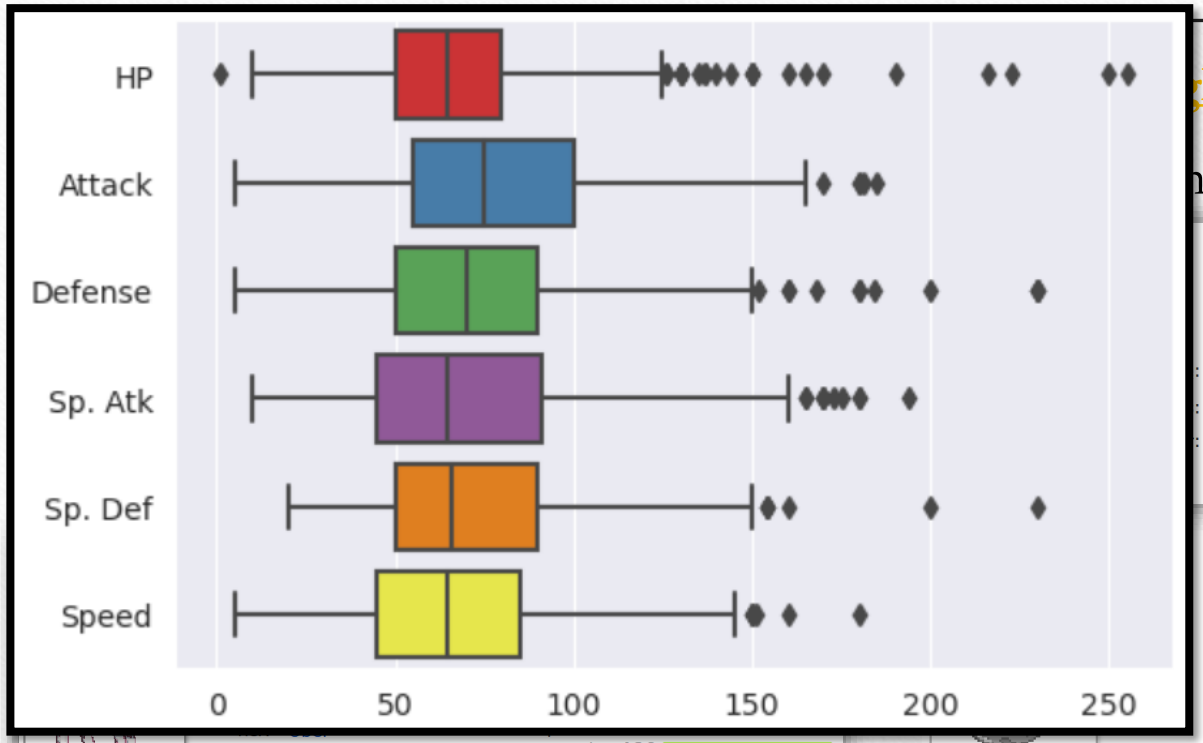
Type: Water
Abilities: Rattled, Swift Swim
Tier: LC

HP:	20
Attack:	10
Defense:	55
Sp. Atk:	15
Sp. Def:	20
Speed:	80

- Special class : **Legendaries**
- Characterized by higher base stats, rare type combinations
- "Outliers"?

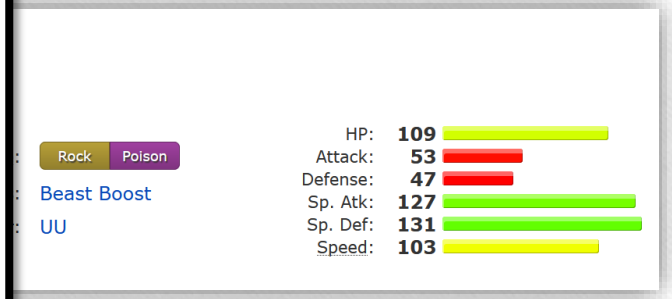
INTRODUCTION

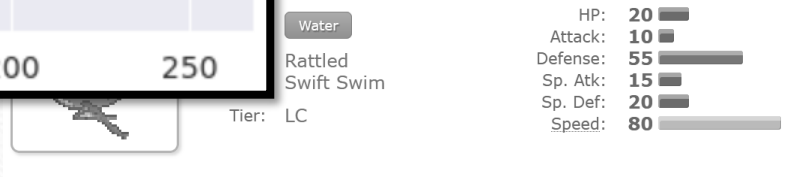
The data of Pokémon



gle.com/rounakbanik/pokemon

nique features

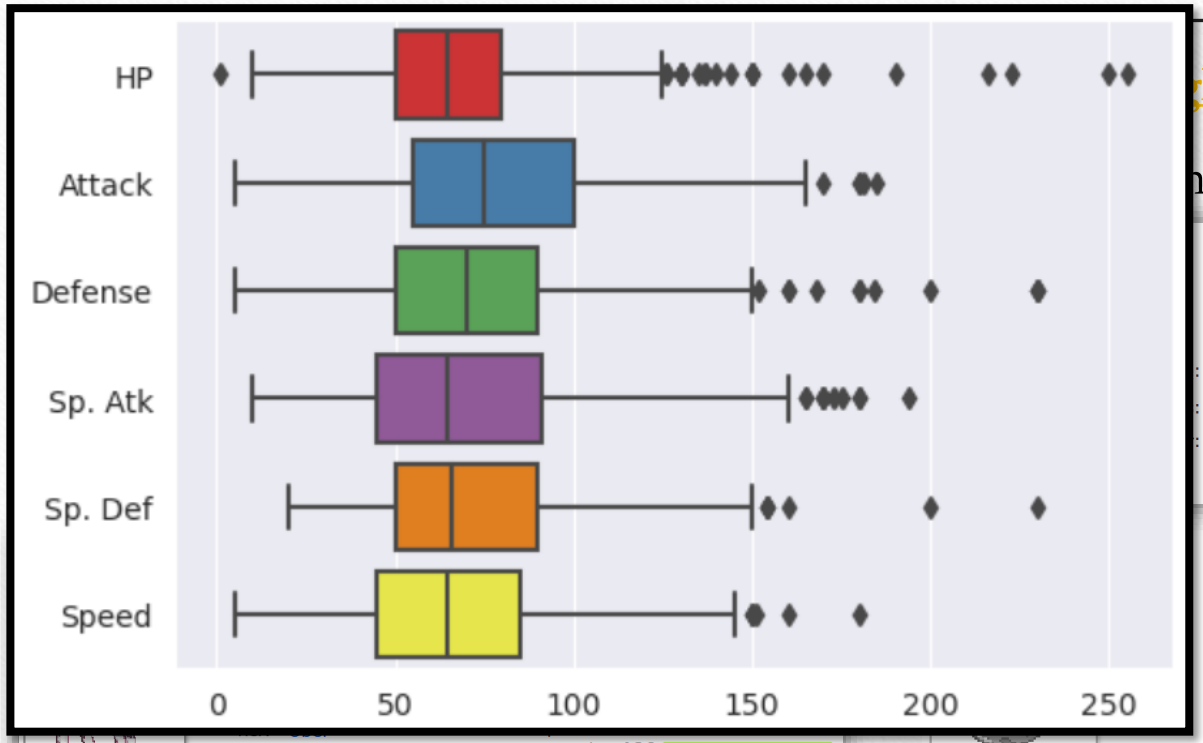




- Special class : **Legendaries**
- Characterized by higher base stats, rare type combinations
- "Outliers"?

INTRODUCTION

The data of Pokémon



gle.com/rounakbanik/pokemon

nique features

The question we seek to answer: How well can we predict that a Pokémon is a **legendary Pokémon?**

combinations

- "Outliers"?

TABLE OF CONTENTS

UNSUPERVISED:

Clustering

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Total	capture_rate	classfic	Defense	experienc	height_m	HP	japanese	name	percentag	#	Sp. Atk	Sp. Def	Speed	Type 1	Type 2	weight_kg	Generatio	Legendary
2	318	45	Seed Po	49	1059860	0.7	45	Fushigida	Bulbasaur	88.1	1	65	65	45	grass	poison	6.9	1	FALSE
3	405	45	Seed Po	63	1059860	1	60	Fushigiso	Ivysaur	88.1	2	80	80	60	grass	poison	13	1	FALSE
4	625	45	Seed Po	123	1059860	2	80	Fushigiba	Venusaur	88.1	3	122	120	80	grass	poison	100	1	FALSE
5	309	45	Lizard P	43	1059860	0.6	39	Hitokage	Charmand	88.1	4	60	50	65	fire		8.5	1	FALSE

- Class ➔ **Legendary** (last column)
- Pairwise clustering on numerical columns
  - Total vs Sp. Atk
  - Capture_rate vs Speed
  - HP vs weight_kg
  - …

➔ $\frac{11!}{2!9!}$ = **55** possible pairwise clusters ➔ unrealistic and waste of time

UNSUPERVISED:

Clustering

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Total	capture_rate	classfic	Defense	experienc	height_m	HP	japanese	name	percentag	#	Sp. Atk	Sp. Def	Speed	Type 1	Type 2	weight_kg	Generatio	Legendary
2	318	45	Seed Po	49										45	grass	poison	6.9	1	FALSE
3	405	45	Seed Po	63										60	grass	poison	13	1	FALSE
4	625	45	Seed Po	123										80	grass	poison	100	1	FALSE
5	309	45	Lizard P	43										65	fire		8.5	1	FALSE

Generally prominent features:

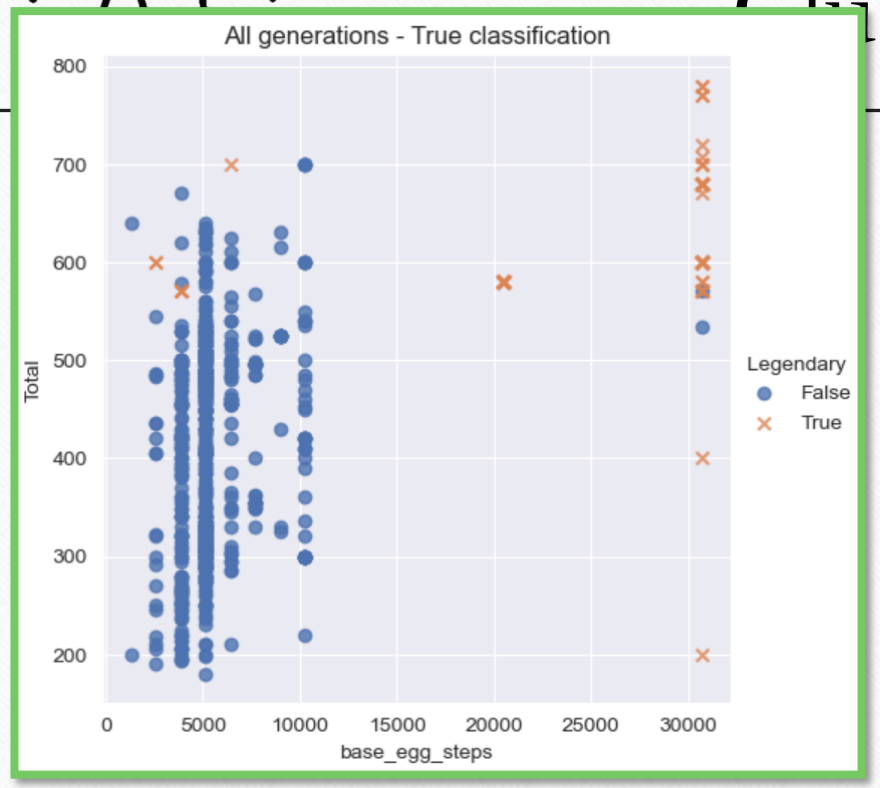
1. base_egg_steps
2. Total
3. base_happiness
4. experience_growth
5. Sp. Atk

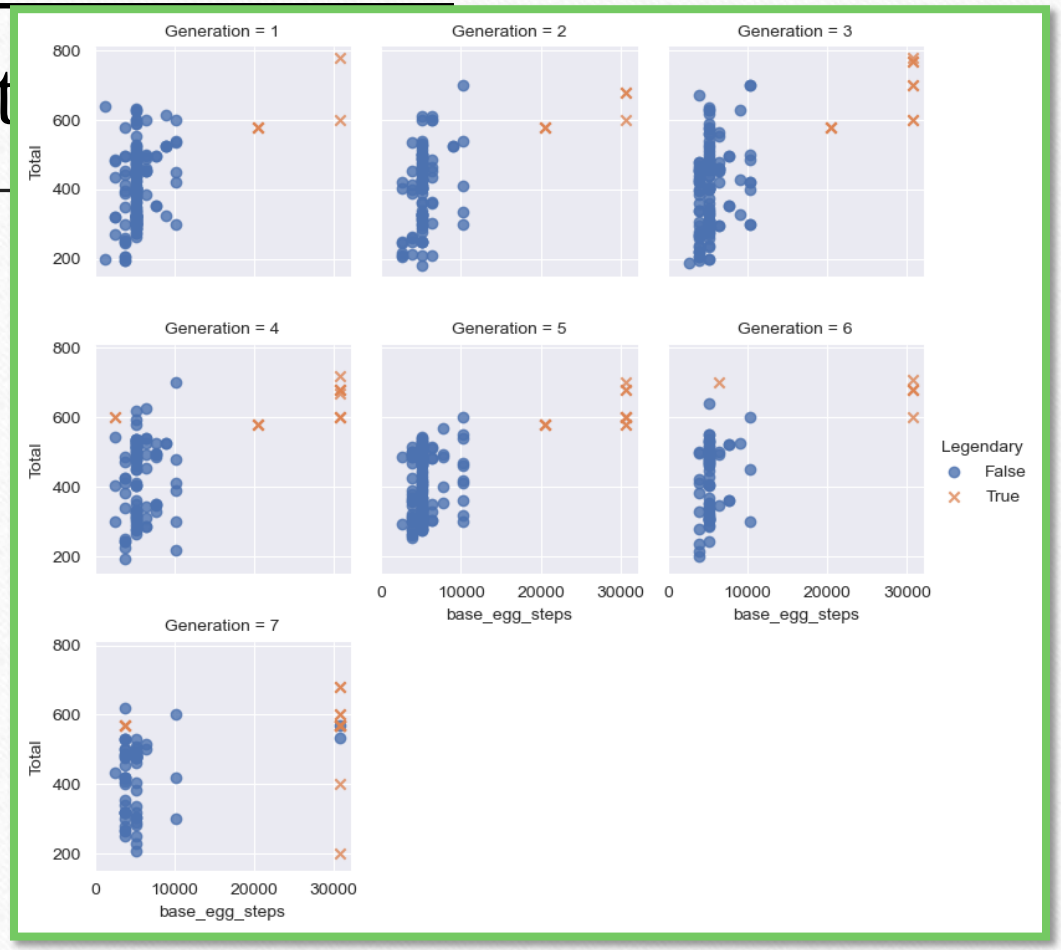
- Class ➔ **Legendary**
- Pairwise clustering on
  - Total vs Sp. Atk
  - Capture_rate vs Sp
  - HP vs weight_kg
  - …
- Feature selection based on **pairwise correlation**

➔ unrealistic and waste of time

UNSUPERVISED:

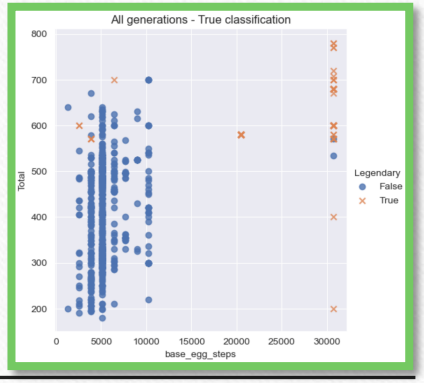
Clust

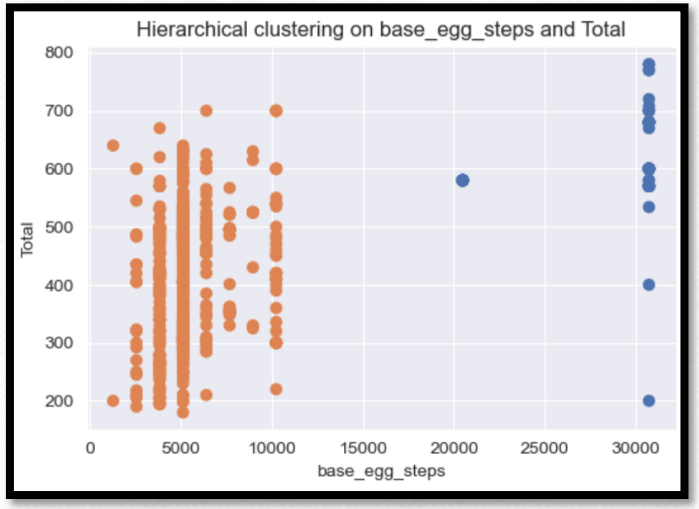


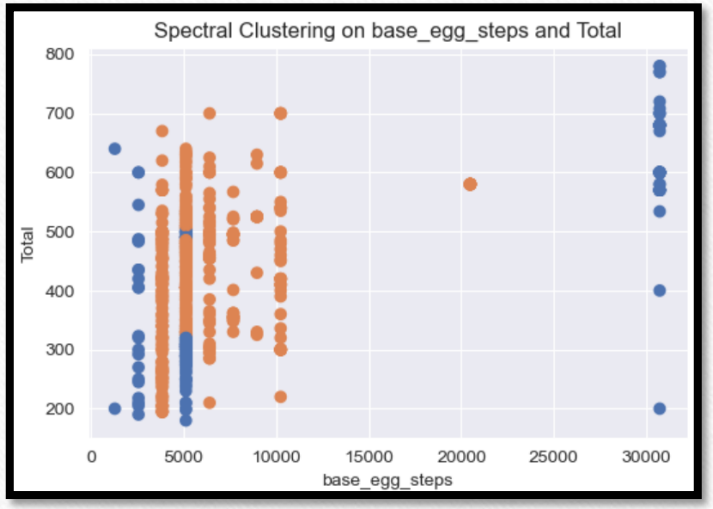


UNSUPERVISED:

Clustering



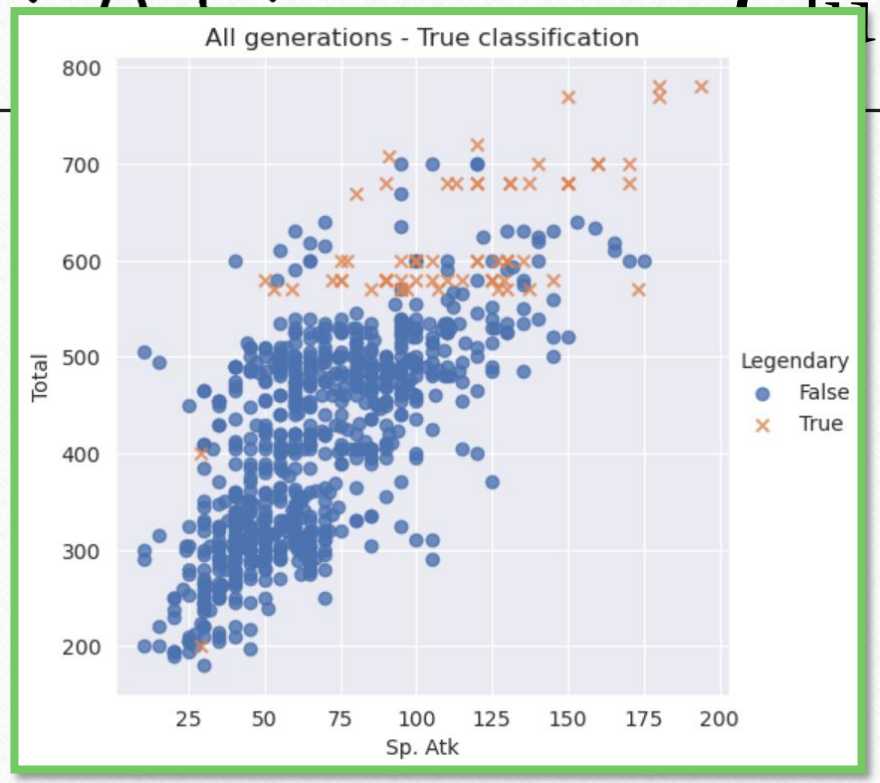


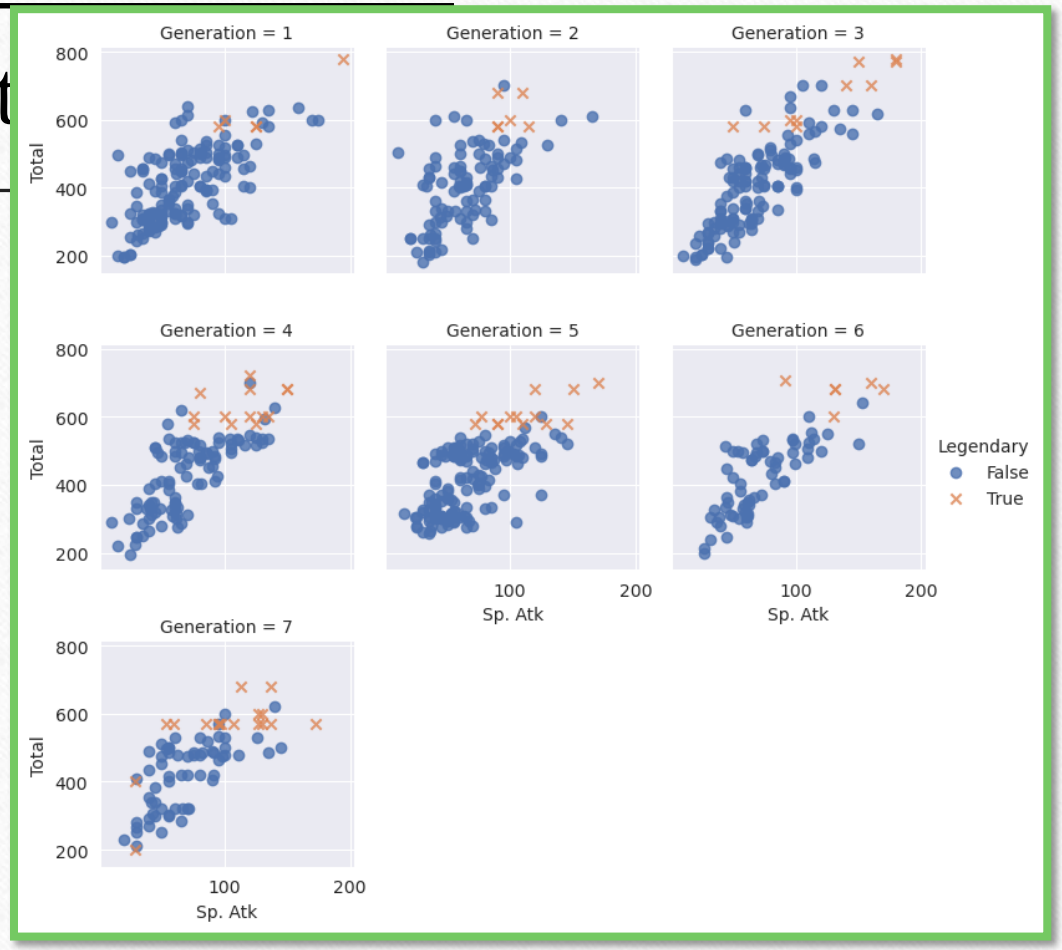


Good results on K-means and hierarchical.
Data is quite easily separable with few outliers.

UNSUPERVISED:

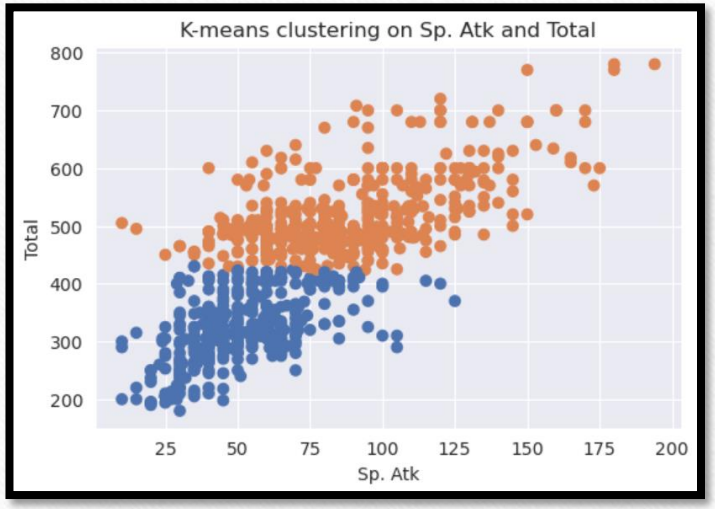
Clust

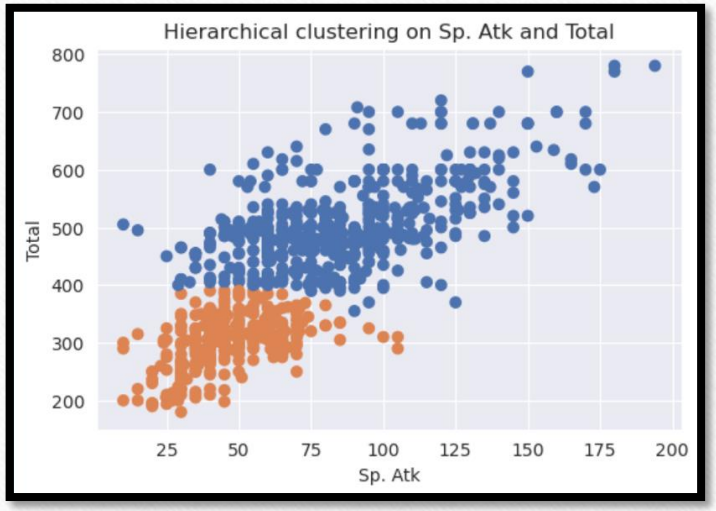


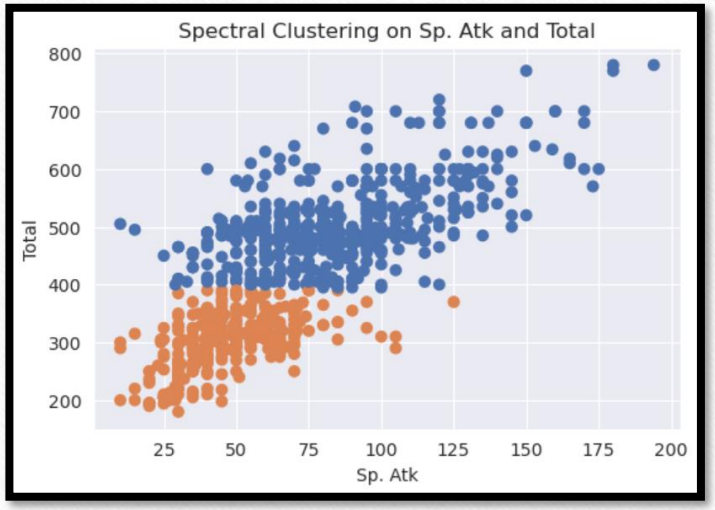


UNSUPERVISED:

Clustering







Bad results all around. Cluster sizes presumed equal, so clustering is inaccurate.

UNSUPERVISED:

Clustering - PCA

- What about integrating **all** features – then reduce dimensions by PCA?
- Kept **all** numerical features, besides Generation, Pokédex number, and non integers

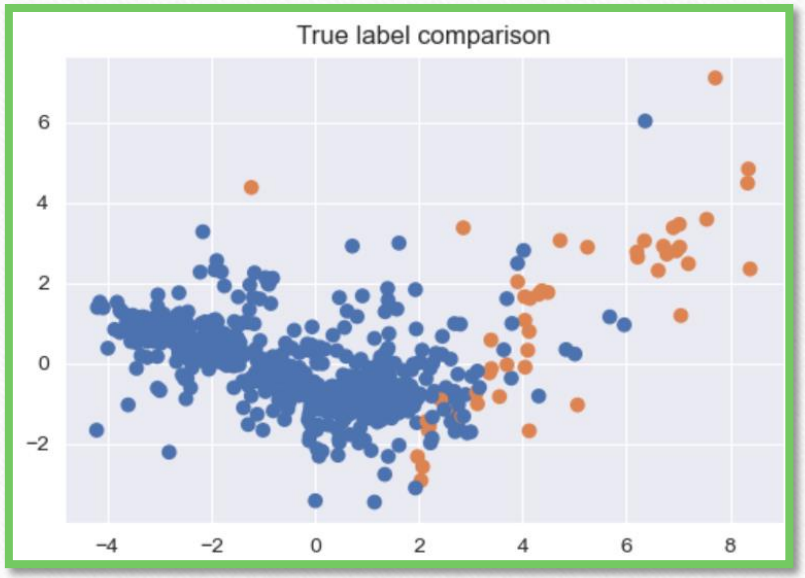
```
### --- PCA STEP --- ###
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

K-means | **Hierarchical** | **Spectral**

UNSUPERVISED:

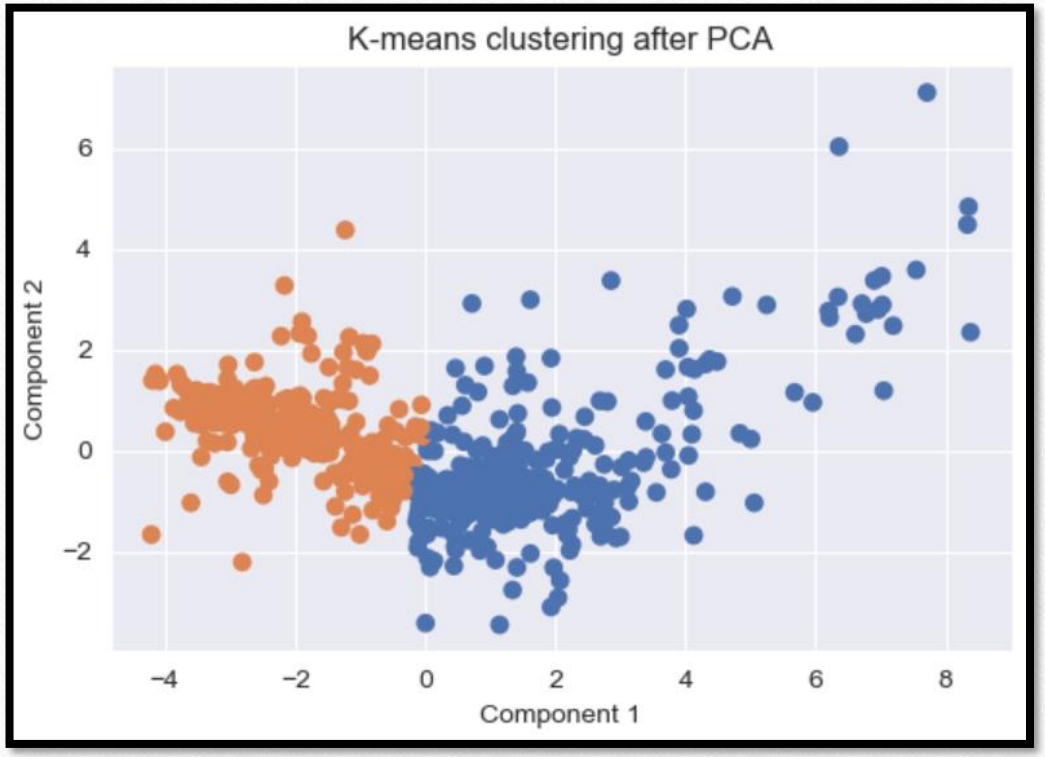
Clustering - PCA

- What about integrating **all** features – then reduce dimensions by PCA?
- Kept **all** numerical features, besides Generation, Pokédex number, and non integers



UNSUPERVISED:

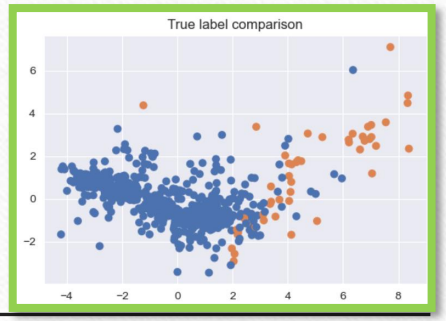
Clustering - PCA

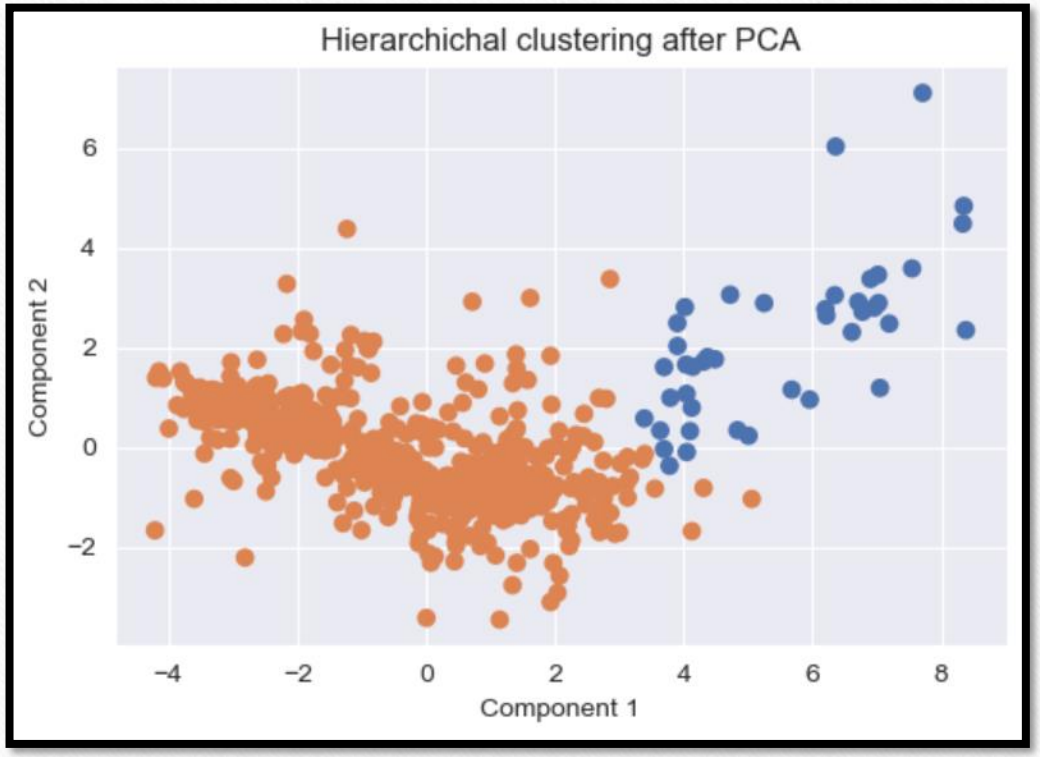


- Homogeneity score: 0.170689700924776 53
- Completeness score: 0.067571837683058 83
- V Measure score: 0.096816438208626 26
- Adjusted rand score: 0.012093226860007383
- Silhouette score: 0.036493747845628 71

UNSUPERVISED:

Clustering - PCA

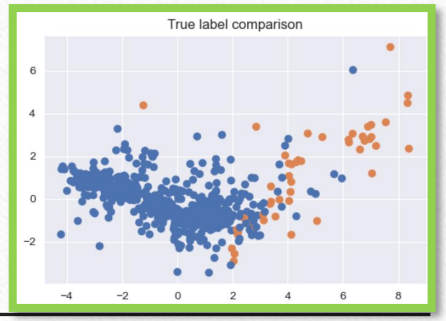


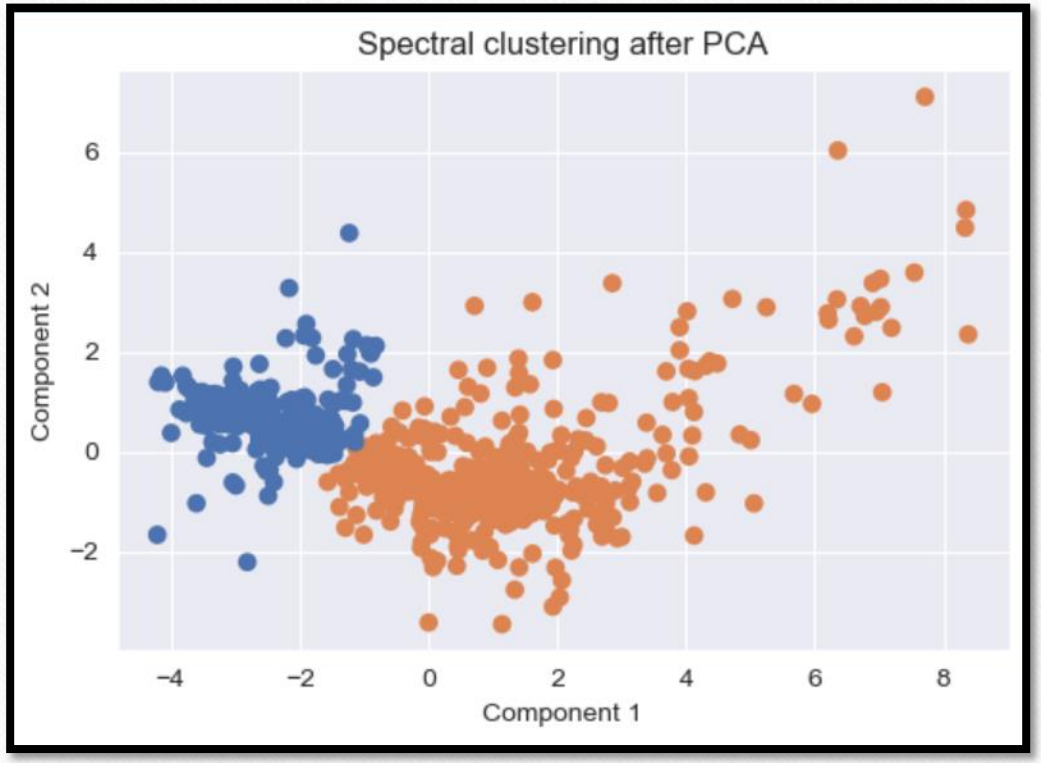


- Homogeneity score: 0.38232846481160215
- Completeness score: 0.43283834067599347
- V Measure score: 0.4060185404710326
- **Adjusted rand score: 0.6136722882689195**
- Silhouette score: 0.37644693014139285

UNSUPERVISED:

Clustering - PCA





- Homogeneity score: 0.10268463671385511
- Completeness score: 0.04361430713995888
- V Measure score: 0.06122421893446118
- Adjusted rand score: -0.04584396494822646
- Silhouette score: -0.02777771761195507

UNSUPERVISED:

Clustering - PCA

Does PCA work?

➔ Hierarchical clustering performs decently

➔ Still a far cry from clustering only "relevant" features

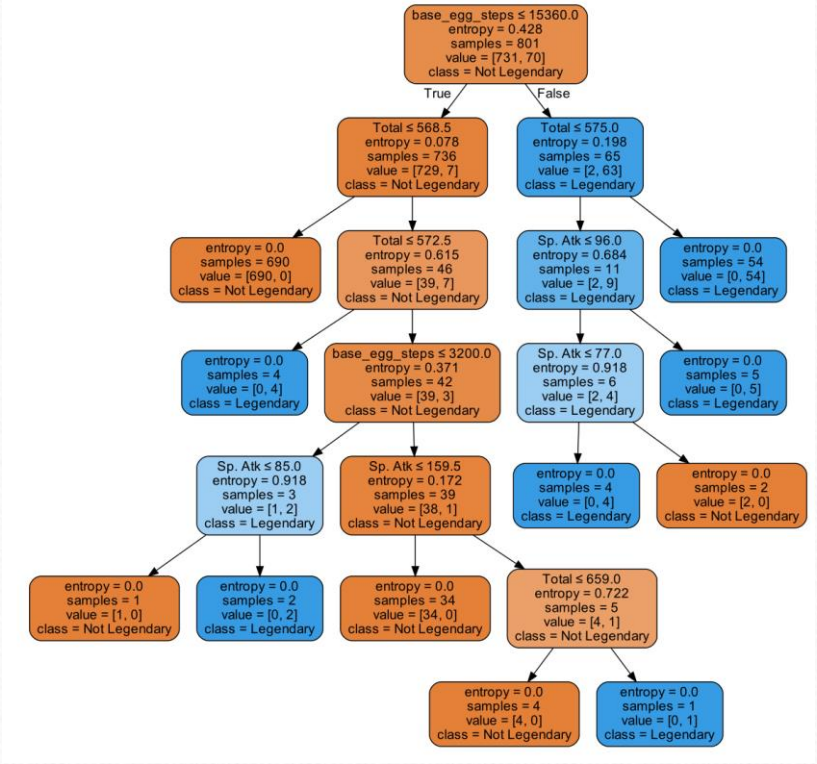
Why?

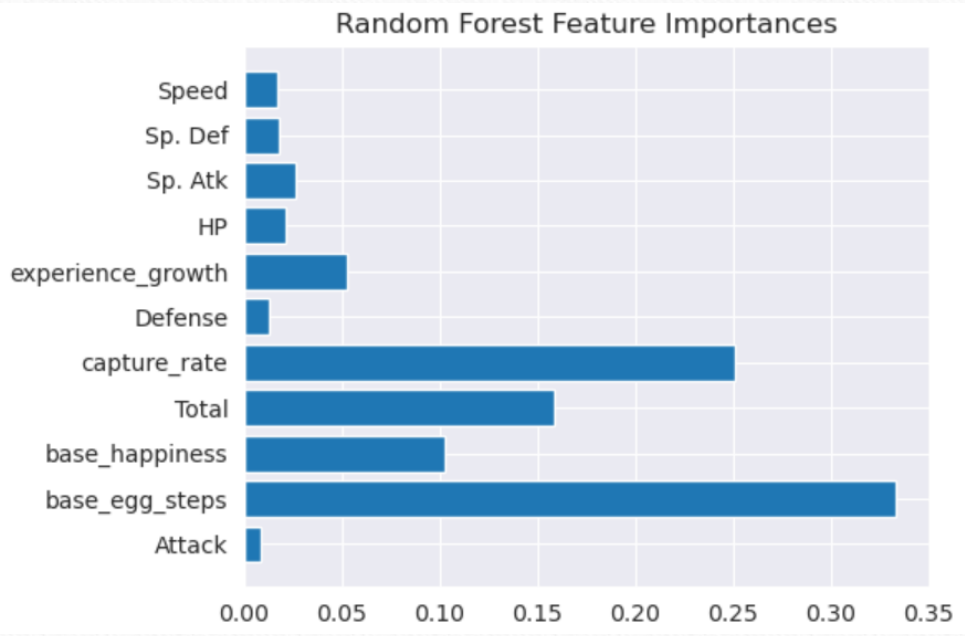
➔ Likely due to certain features being much more **important** (higher weight)

TABLE OF CONTENTS

SUPERVISED:

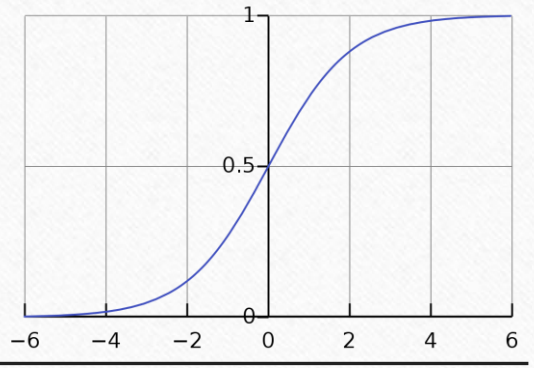
Decision tree + Forest





SUPERVISED:

Logistic regression



- Logistic Regression disappointing
  - High accuracy but low precision
  - Feature weights do not correspond to feature correlation

```
Average accuracy : 0.9063540372670807
Average precision : 0.2316011904761905
Average Theta : [-0.42456532  0.23226571 -0.30397868 -0.55889356]
```

- **New idea**: directly link the correlation and the predictions

```
[0.48543982279986336, 0.873488340399458, -0.4131077637494285, 0.361038087952768 37]
```

SUPERVISED:

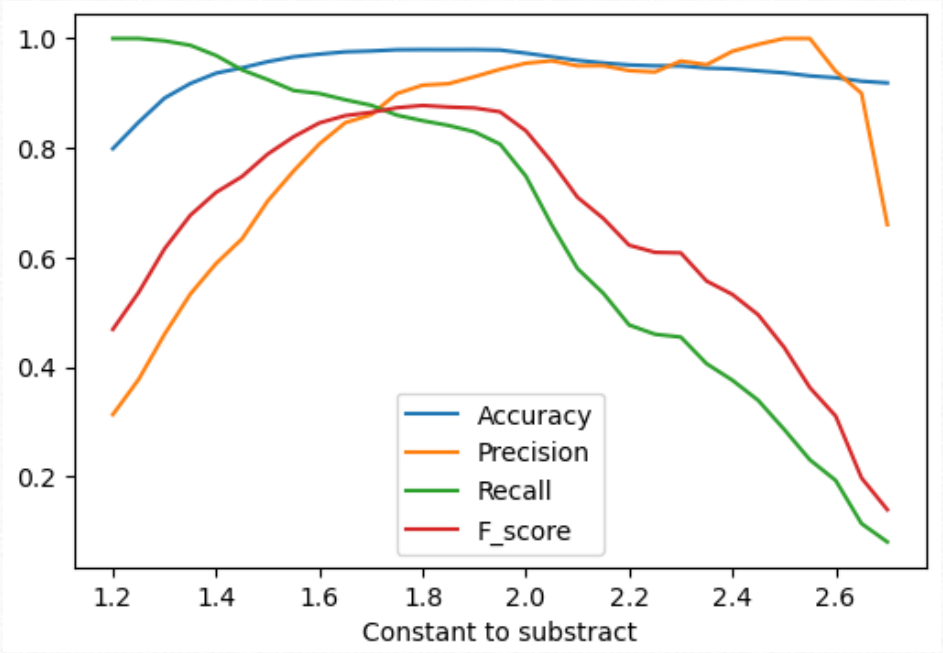
Correlation based method

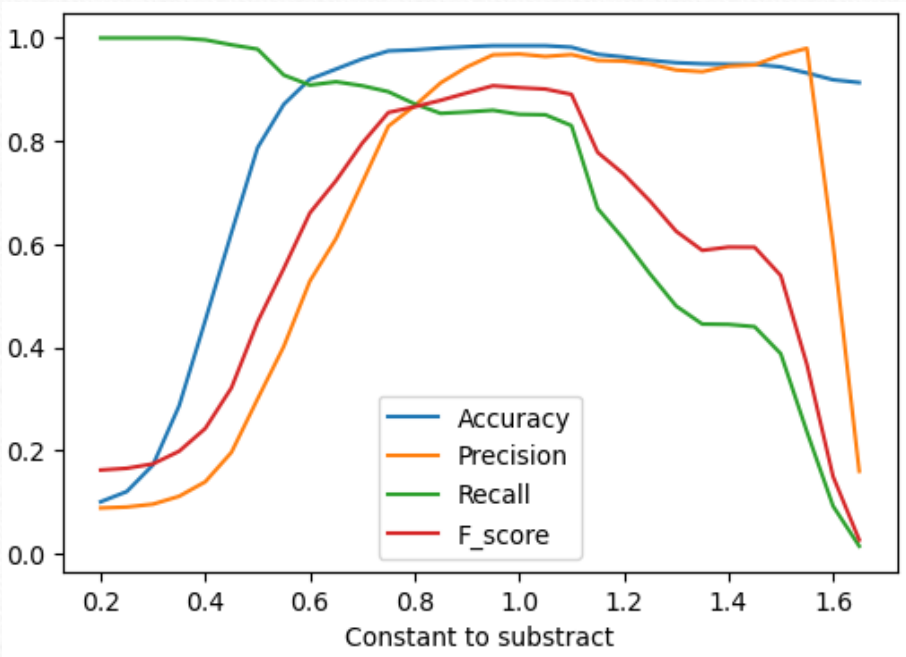
- Weight of feature = correlation with legendary classification Y
- Matrix product of **weights and features**
  - ➔ single value output per feature
    - If greater than threshold: **Legendary**
    - If not: Common

How to calculate the best threshold value?
➔ k-fold cross-validation

SUPERVISED:

Correlation based method



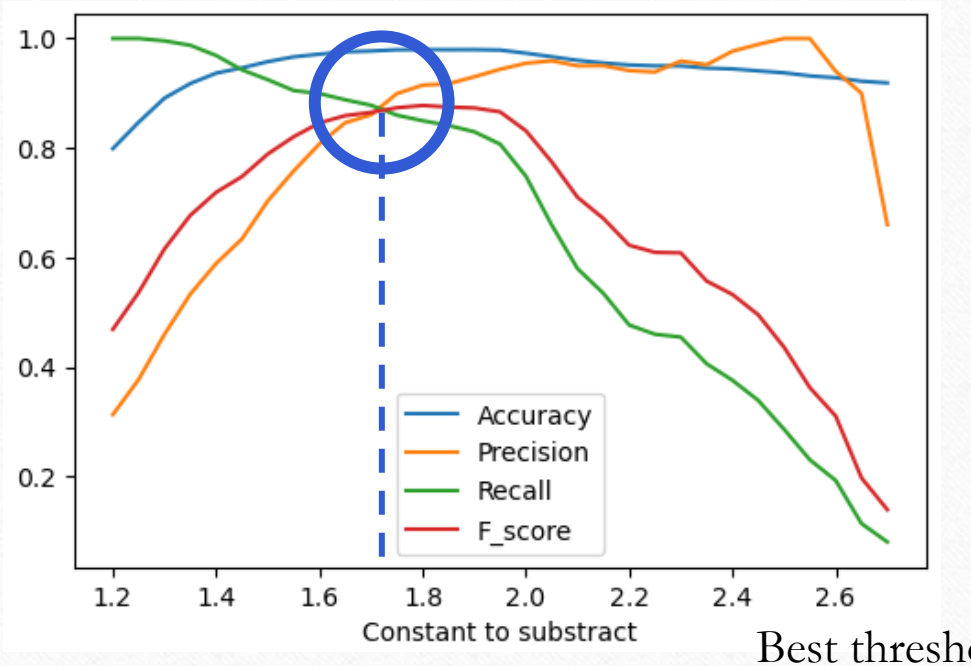


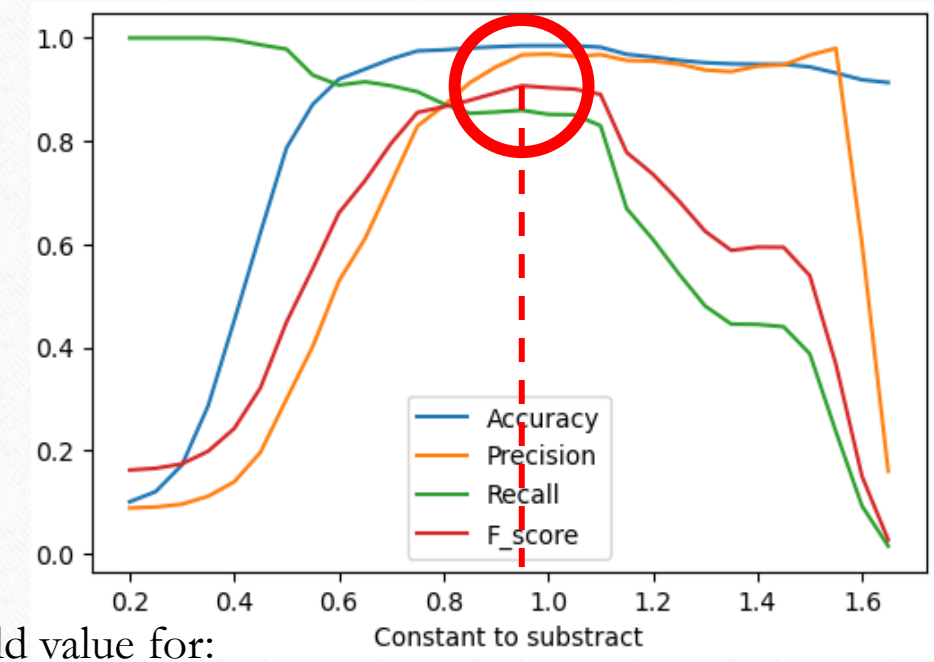
Metrics on **all** features (left)

Metrics on **4 best** features (right)

SUPERVISED:

Correlation based method





Best threshold value for:

All features: left ➔ 1.75

4 best features: right ➔ 0.95

TABLE OF CONTENTS

DISCUSSION

- Dataset is small – only a dozen of features ➔ testing and training was limited.
- **Unsupervised**:
  - Pairwise clustering: **effective** once most prominent features identified
  - All features + PCA: not irrelevant, but not as effective
- **Supervised**:
  - Decision tree: feature importance **confirms clustering conclusions**
  - Logistic regression: poor results
  - Correlation weight method: **good results** after k-fold cross validation

DISCUSSION

❖ Access? ➔ github.com/LiamLeGoffic/SPLEX_Project

Thank you for listening!