

Aprendizado de Máquina

O desafio de duas décadas

Anderson Faria (11053613)
Leonardo Nascimento (11051613)



tiBiA

Detectando o uso de ferramentas de automação (BOTs)



~~União~~

Por quê?



tibia

Relevância

Forum statistics

Threads: 212,432
Messages: 2,184,728
Members: 223,876

STATISTICS

There are **10165** players online on **373** servers
Last Update: April 28, 2019, 11:48 am

We have **495** servers in our database
Current Time: April 28, 2019, 11:48 am

📊 Estatísticas Gerais

88797

Total Topics

494961

Total Posts

193678

Total Members

1133

Most Online



forgotten server



Repositories

37

Code

4M

Commits

47K

Issues

21K

Marketplace

0

Topics

1

Wikis

1K

Users

0

Brasileiro e 'pirata', game online de "Pokémon" já tem meio milhão de fãs



Dragon Ball Online (DBO)



Digiwo



Grande Line Adventures (GLA)



Narutibia (NTO)



Bloodstone Online



Avatar Online





Sem solução até hoje

Detecção Manual



Limitação de funcionalidades



Tentativa de confusão

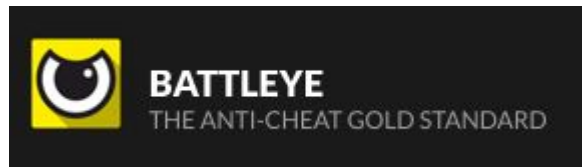


Passando a bola: 1ª vitória



MASS DELETE: 11 MIL CONTAS PARA DELETERA!

Ajuda dos universitários



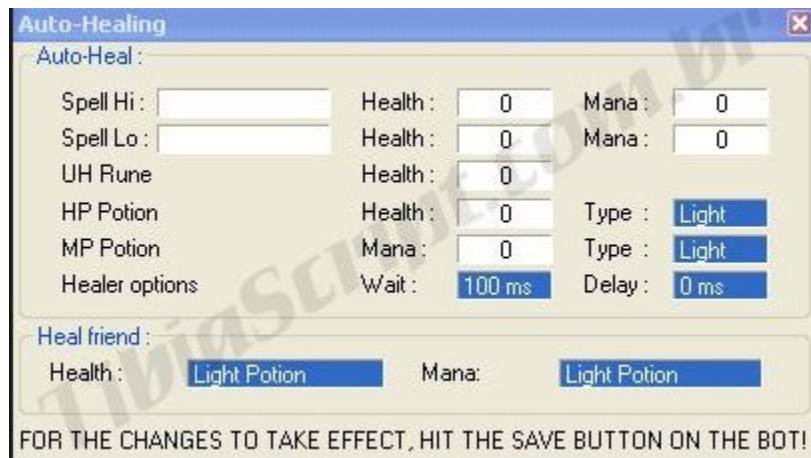
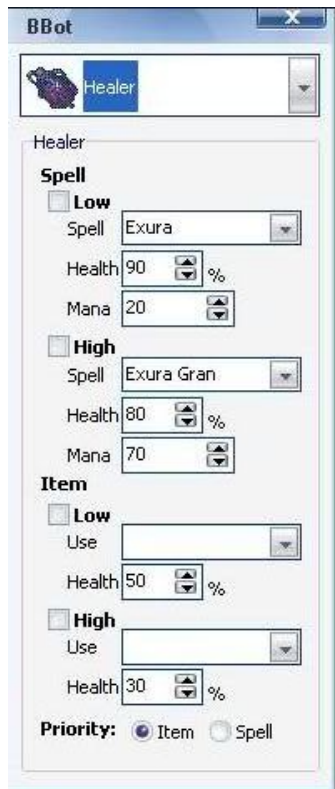
Conhecendo o inimigo

Se você conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas. Se você se conhece mas não conhece o inimigo, para cada vitória ganha sofrerá também uma derrota. Se você não conhece nem o inimigo nem a si mesmo, perderá todas as batalhas...

Sun Tzu



Conhecendo o inimigo



Conhecendo o inimigo

The image shows a screenshot of the Cavebot configuration window, a software interface for configuring a bot. The window is divided into several sections with various settings and controls.

Healing/Protection-Spell and/or Hotkey

Light :	exura	%	80	MP	20
Heavy :	exura gran	%	60	MP	70
Mana Potion MP%	80	Potion type	gma		
Heal Potion HP%	0	Potion type	--		

Editable Hotkeys

Insert: say exani hu Save settings

End: SDMAX Home: say exani hur

PageD: amul 3158 PageU: say utevo re

Num0: F11 Num1: say exori vis

Num2: ring 3510 Num4: ANTIPUSH

Num7: say exevo g. Num(): FOLLOW

Magelomb

Log On and Start Stop

Timers

Fire Timer: 300 Loot Timer: 300

Heal Timer: 60 Boost Timer: 10

Team Server

127.0.0.1 password Connect

SD ☒ Potion Friend ☐ I am Leader

Vip Check

IP Change

GO

Help/Info

Creaturelist

Bot heals all players in VIP marked with a heart using EXURA SID

Cavebot

Start Record Stack Loot in Backpack Range to attack Monster 3

Stop Save Stack Loot on Ground

Load Loot Before Kill MF/Ammo Reloader BP #

If Player or GM Enter

Say: Beep Attack Mode: Dodge/Hotkey? Pause Hotkey?:

Loot Manager

☒ Auto-Loot ☐ AutoOpenCorse ☐ Seller

Sort Loot into 20BP Sort Loot to Ground Load Loot from Ground Buy Mana

PvP/ Heal Friend

☐ AutoFireVipSkull ☐ AutoFireNonSafeList ☐ Enemy Colors

☒ AutoSio-Hearts ☐ AutoUH-Hearts Heal Vip Hearts % 60

Extras

☐ Anti-Idle ☐ RuneMake ☐ Camouflage

☐ StepBack ☐ Autolog ☐ Hold Attack ☒ Level Spy

Specialty

☐ AutoSSA/Mring ☒ AutoManashield ☒ AutoGranHur

☒ Click-Reuse ☐ MC Active 1 ☐ Always UH

☐ Speedboost ☐ AutoSkin ☐ AutoEat ☐ UH if No Mana

Remote Control

Control Character Alerts Remote Control

Combo UE?

License #

Enter Key Here

Load #

Funções mais comuns

Coleta automática



Auto cura



Automação de caça (cave bot)



Metodologia

Coletar dados e criar uma base a partir deles

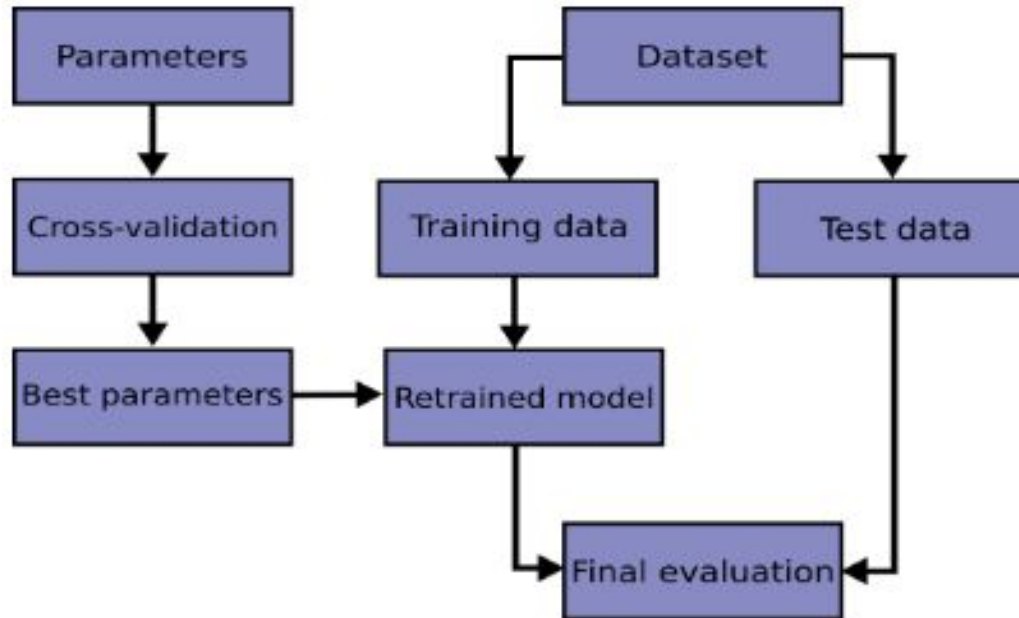
```
class AntibotDataset:
    def __init__(self, total_players, bot_rate, seed=None):
        self.total_players = total_players
        self.bot_rate = bot_rate
        np.random.seed(seed)

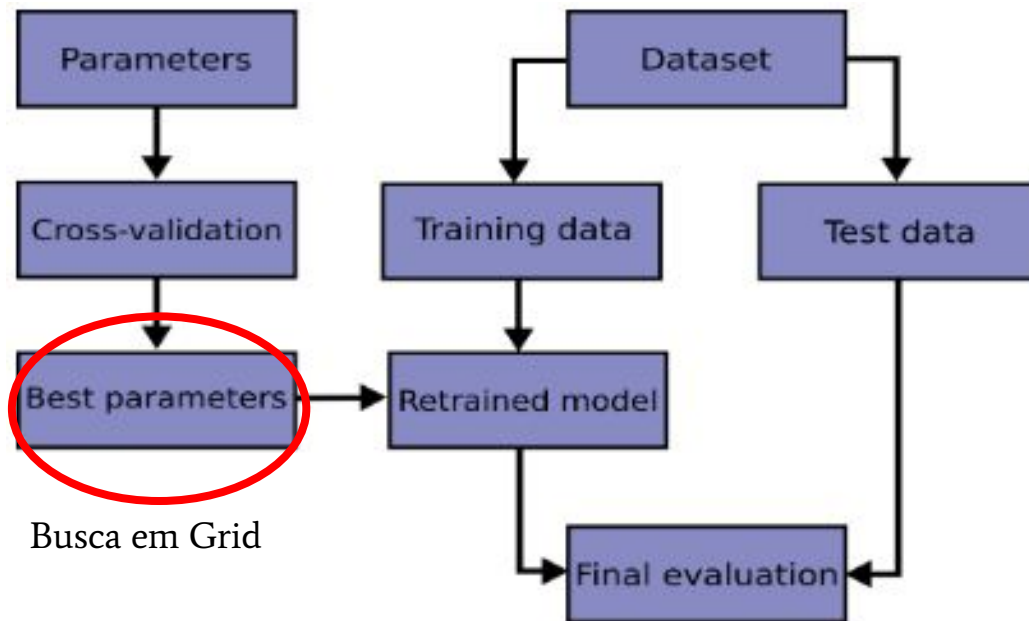
    def generate(self):
        # Quantidade de jogadores bot na amostra
        self.bot_players = round(self.total_players * self.bot_rate)
        # Quantidade de jogadores reais na amostra
        self.real_players = self.total_players - self.bot_players

        columns = (
            self.collected_items(),
            self.avg_time_to_collect_item(),
            self.delta_time_to_collect_item(),
            self.reaction_to_heal(),
            self.avg_reaction_time_to_heal(),
            self.delta_reaction_time_to_heal(),
            self.killed_enemies(),
            self.hungry()
        )

        self.data = np.column_stack(columns)
```

Fluxo de execução





Comparando clasificadores

Classificador	Accuracy	Precision	F-Measure	Recall	ROC AUC
Decision Tree	0.994 +/- 0.008	0.981 +/- 0.031	0.985 +/- 0.020	0.990 +/- 0.030	0.997 +/- 0.004
SVC Poly	0.992 +/- 0.010	0.981 +/- 0.032	0.980 +/- 0.024	0.980 +/- 0.024	0.999 +/- 0.002
SVC RBF	0.992 +/- 0.010	0.990 +/- 0.020	0.979 +/- 0.025	0.970 +/- 0.040	0.998 +/- 0.005
Linear SVC	0.991 +/- 0.011	0.985 +/- 0.032	0.977 +/- 0.028	0.970 +/- 0.033	0.999 +/- 0.003
SVC Linear	0.990 +/- 0.010	0.985 +/- 0.023	0.974 +/- 0.026	0.965 +/- 0.039	0.999 +/- 0.003
K-Neighbors	0.986 +/- 0.014	0.981 +/- 0.038	0.964 +/- 0.035	0.950 +/- 0.055	0.995 +/- 0.011

Candidatos

Classificador	Accuracy	Precision	F-Measure	Recall	ROC AUC
Decision Tree	0.994 +/- 0.008	0.981 +/- 0.031	0.985 +/- 0.020	0.990 +/- 0.030	0.997 +/- 0.004
SVC Poly	0.992 +/- 0.010	0.981 +/- 0.032	0.980 +/- 0.024	0.980 +/- 0.024	0.999 +/- 0.002
SVC RBF	0.992 +/- 0.010	0.990 +/- 0.020	0.979 +/- 0.025	0.970 +/- 0.040	0.998 +/- 0.005
Linear SVC	0.991 +/- 0.011	0.985 +/- 0.032	0.977 +/- 0.028	0.970 +/- 0.033	0.999 +/- 0.003
SVC Linear	0.990 +/- 0.010	0.985 +/- 0.023	0.974 +/- 0.026	0.965 +/- 0.039	0.999 +/- 0.003
K-Neighbors	0.986 +/- 0.014	0.981 +/- 0.038	0.964 +/- 0.035	0.950 +/- 0.055	0.995 +/- 0.011

Decision Tree

Accuracy	Precision	F-Measure	Recall	ROC AUC
0.994 +/- 0.008	0.981 +/- 0.031	0.985 +/- 0.020	0.990 +/- 0.030	0.997 +/- 0.004

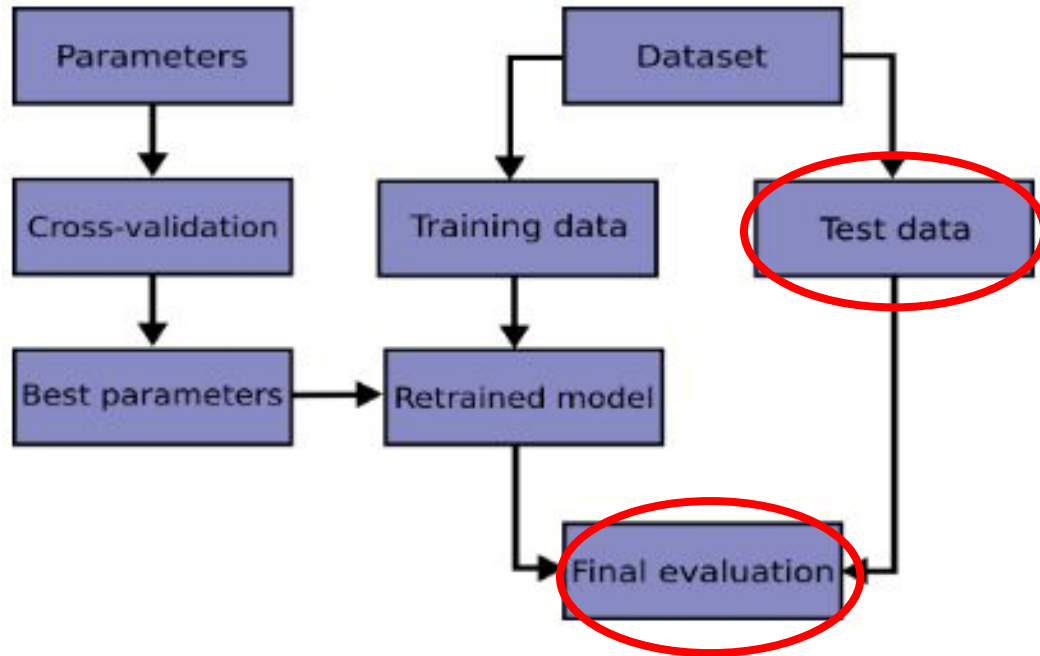
- Melhor acurácia
- Média da área da curva ROC ligeiramente menor
- Ambos com baixo erro
- Taxa de recall mais baixa
- Bônus: + Rápido

SVC (poly)

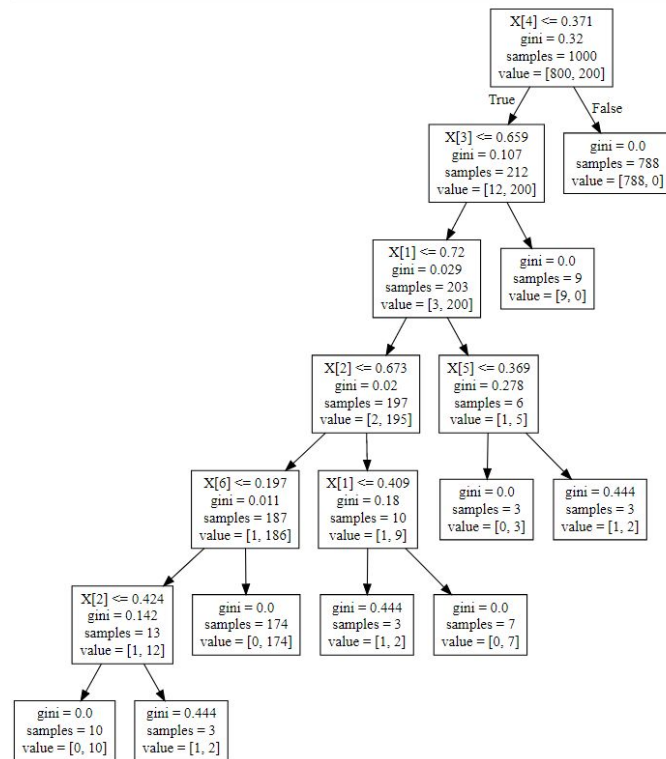
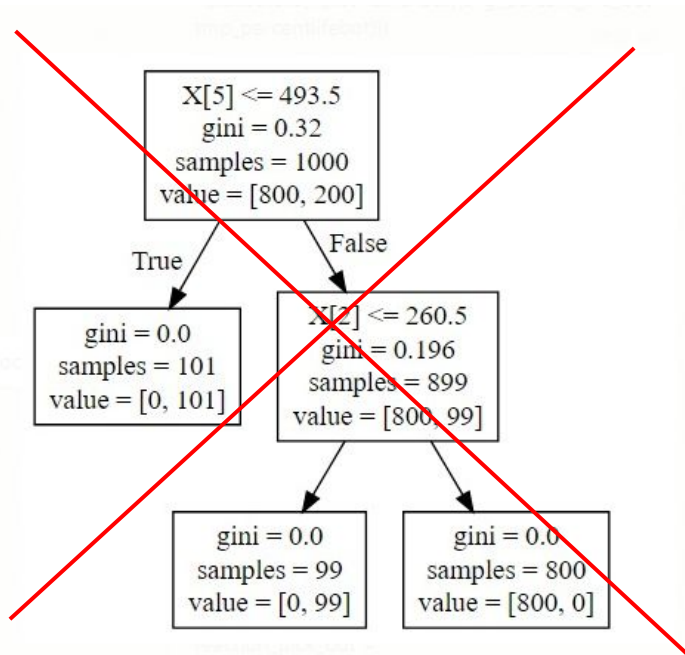
Accuracy	Precision	F-Measure	Recall	ROC AUC
0.992 +/- 0.010	0.981 +/- 0.032	0.980 +/- 0.024	0.980 +/- 0.024	0.999 +/- 0.002

- Segunda melhor acurácia
- Média da área da curva ROC quase 1
- Ambos com baixo erro
- Segunda taxa de recall mais baixa

Fluxo de execução



Gerando a árvore de decisão



Resultado final

```
DecisionTreeClassifier(criterion='gini', min_samples_leaf=3, min_samples_split=2, splitter='best')
  accuracy: 0.994 +/- 0.008
  precision: 0.981 +/- 0.031
  f1: 0.985 +/- 0.020
  recall: 0.990 +/- 0.030
  roc_auc: 0.997 +/- 0.004
```

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	900
1.0	0.87	0.99	0.93	100
micro avg	0.98	0.98	0.98	1000
macro avg	0.93	0.99	0.96	1000
weighted avg	0.99	0.98	0.98	1000

Resultado final

```
SVC(C=3.0, degree=2, gamma='scale', kernel='poly')
```

```
accuracy: 0.992 +/- 0.010
```

```
precision: 0.981 +/- 0.032
```

```
f1: 0.980 +/- 0.024
```

```
recall: 0.980 +/- 0.024
```

```
roc_auc: 0.999 +/- 0.002
```

	precision	recall	f1-score	support
0.0	1.00	0.99	0.99	900
1.0	0.90	0.97	0.93	100
micro avg	0.99	0.99	0.99	1000
macro avg	0.95	0.98	0.96	1000
weighted avg	0.99	0.99	0.99	1000

```
[222 258 262 297 336 481 493 514 599 622 625 687 697 720 757 887 904 906  
929 1 4 6 57 72 123 133 152 158 159 170 179 186 187 198 199]
```

```
Number of support vectors for each class
```

```
[19 16]
```

Resultado final



Dúvidas?