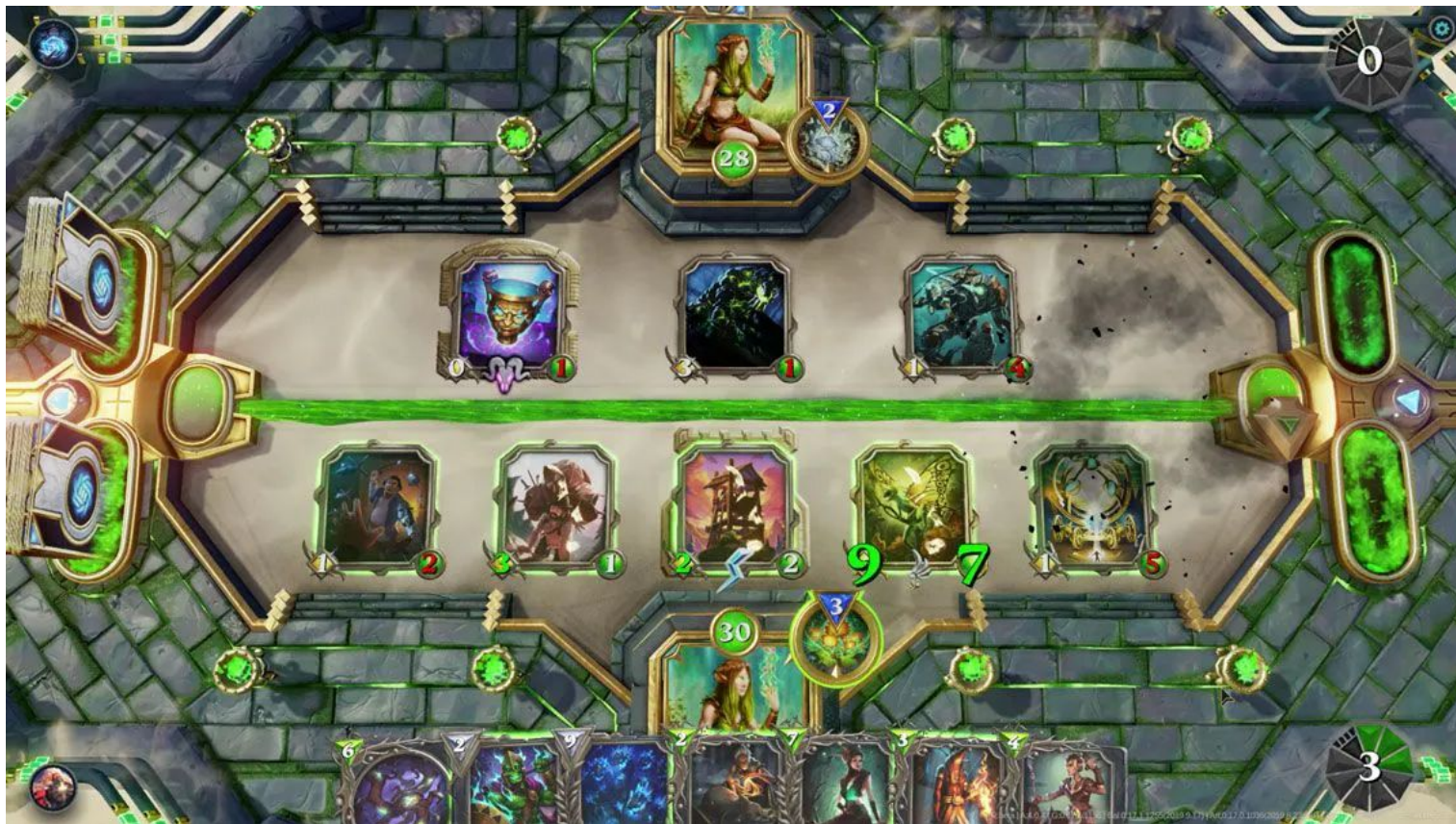


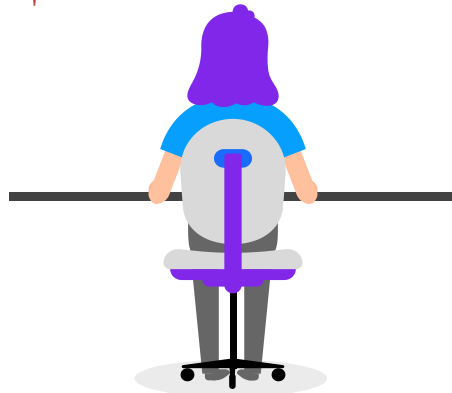
Gods Unchained

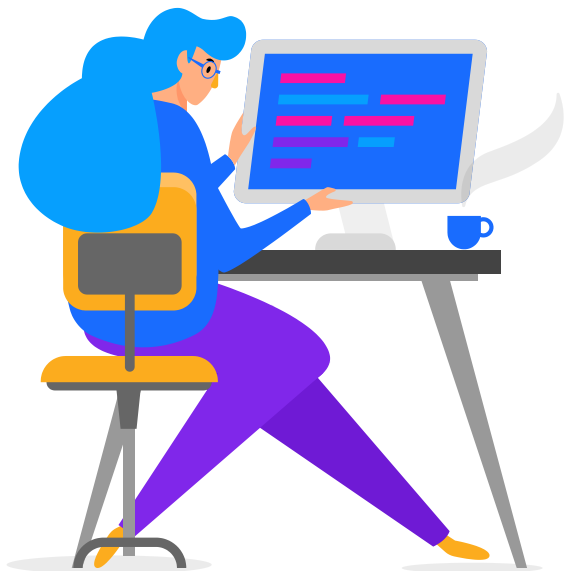
Lucas de Brito Silva

O que é Gods Unchained?



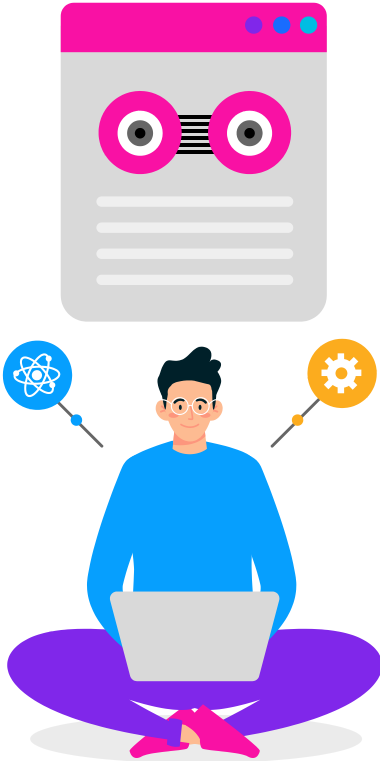
— GODS — UNCHAINED





O problema é saber
qual é o momento certo de se
lançar determinada carta.

Sumário



01

EDA

02

Construção da
Pipeline

03

API

04

Melhorias e
Aprendizado

Exploratory Data Analysis

Preprocessing



Dataset de Treinamento

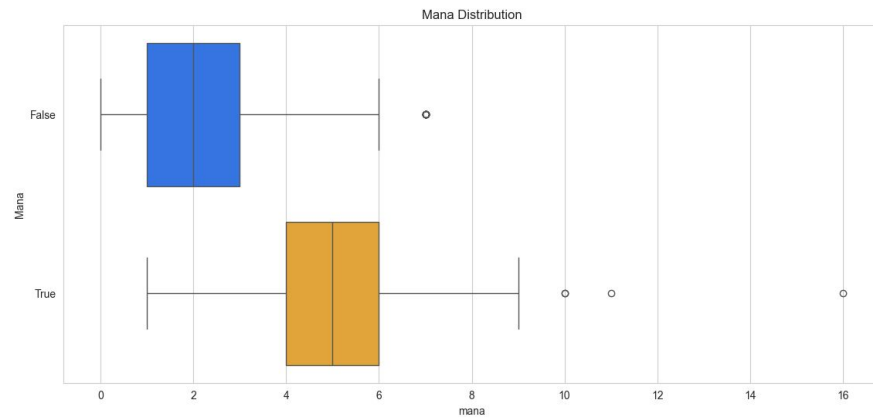
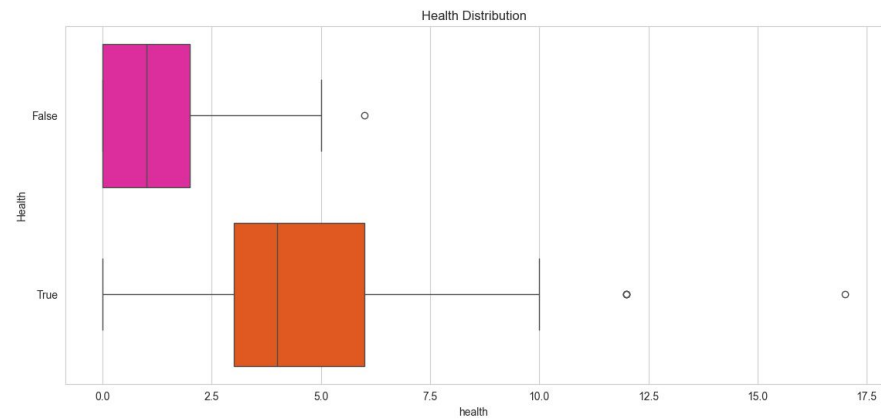
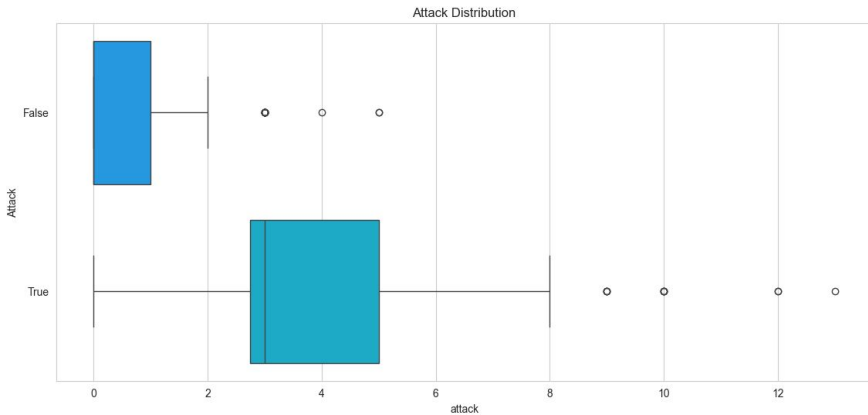
- *Dataset* com **788** linhas e **8** colunas;
- *Dataset* sem valores faltantes;
- **432** cartas classificadas como “*early*” e **356** classificadas como “*late*”;
- A coluna de variável resposta (*strategy*) pode ser transformada para booleano, ou seja, 1 ou 0.

	id	name	mana	attack	health	type	god	strategy
0	1118	Firewine	5	0	0	spell	nature	early
1	1036	Leyhoard Hatchling	10	2	1	creature	magic	late
2	244	Aetherfuel Alchemist	6	4	4	creature	neutral	late
3	215	Millenium Matryoshka	4	2	2	creature	neutral	late
4	87013	Poison Peddler	4	1	3	creature	neutral	late

```
id          int64
name        object
mana        int64
attack      int64
health      int64
type        object
god         object
strategy    object
dtype: object
```

Exploratory Data Analysis

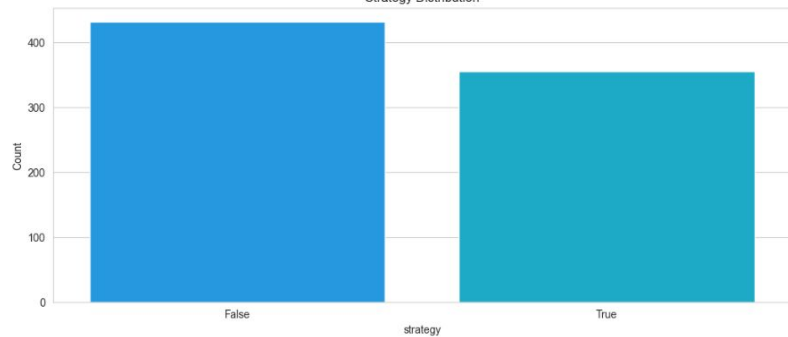
Analysis



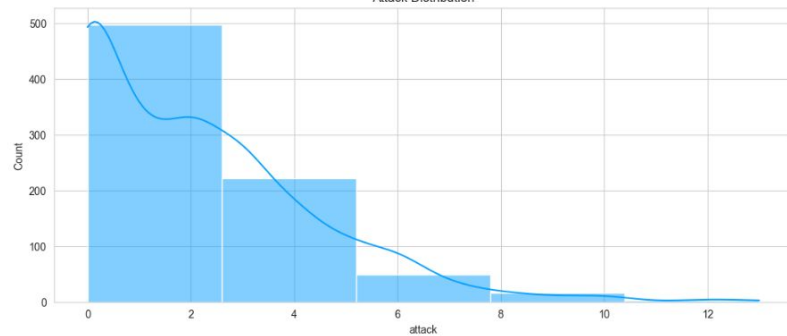
Exploratory Data Analysis

Analysis

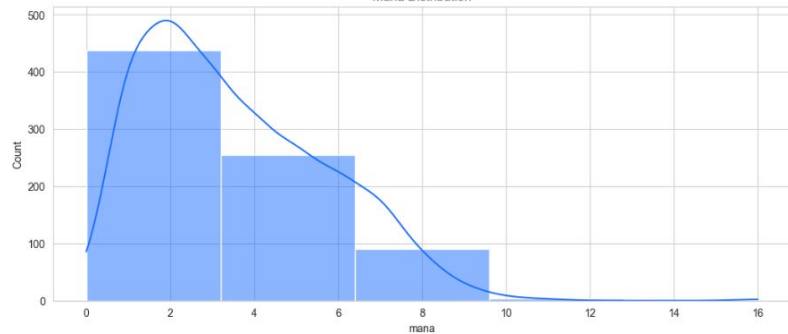
Strategy Distribution



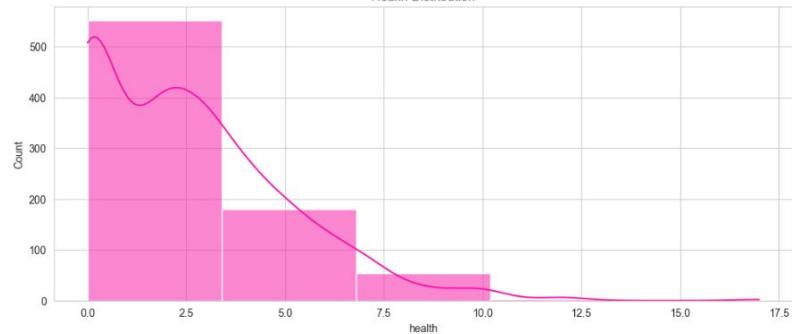
Attack Distribution



Mana Distribution

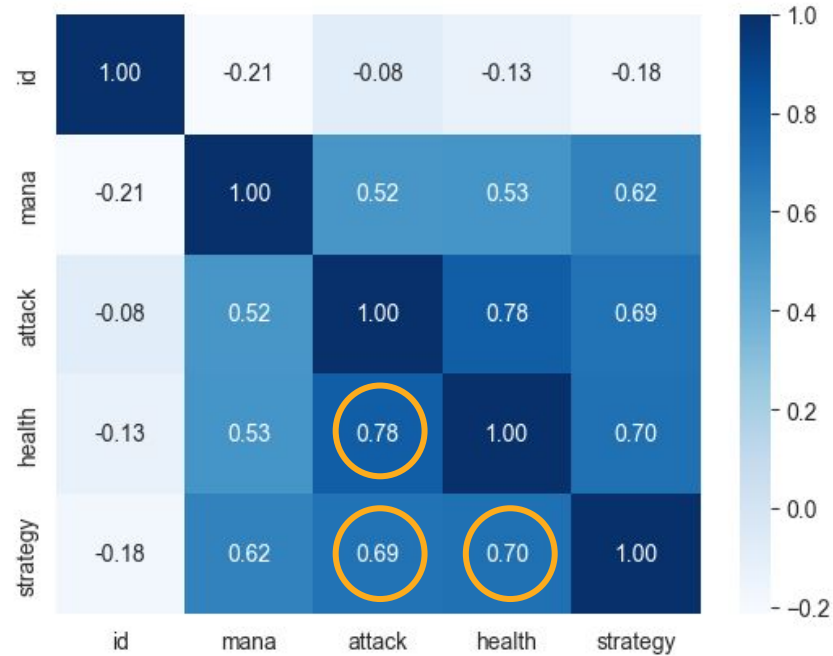


Health Distribution



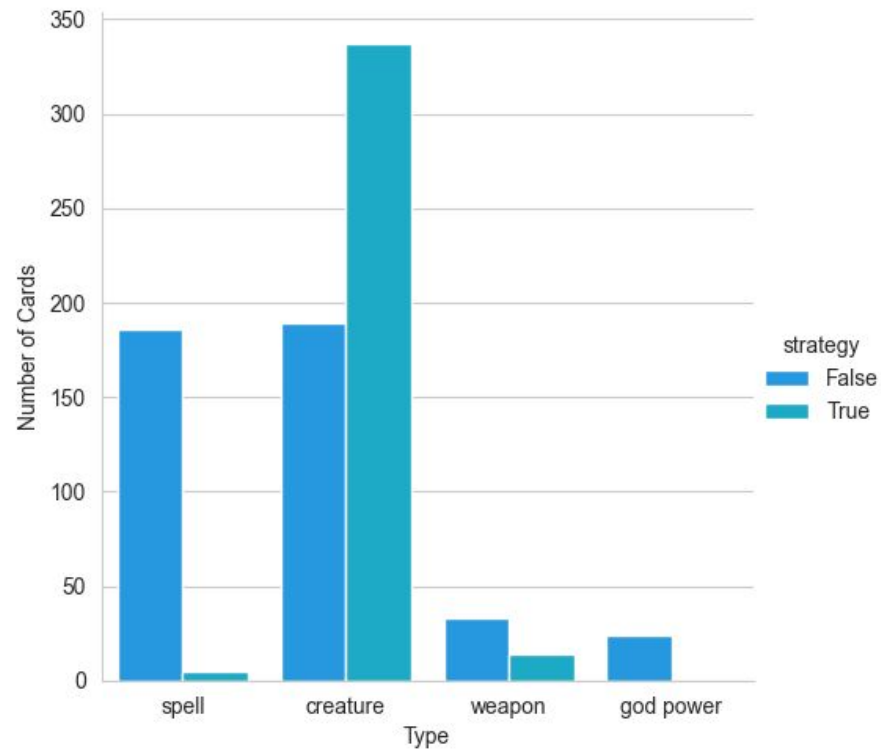
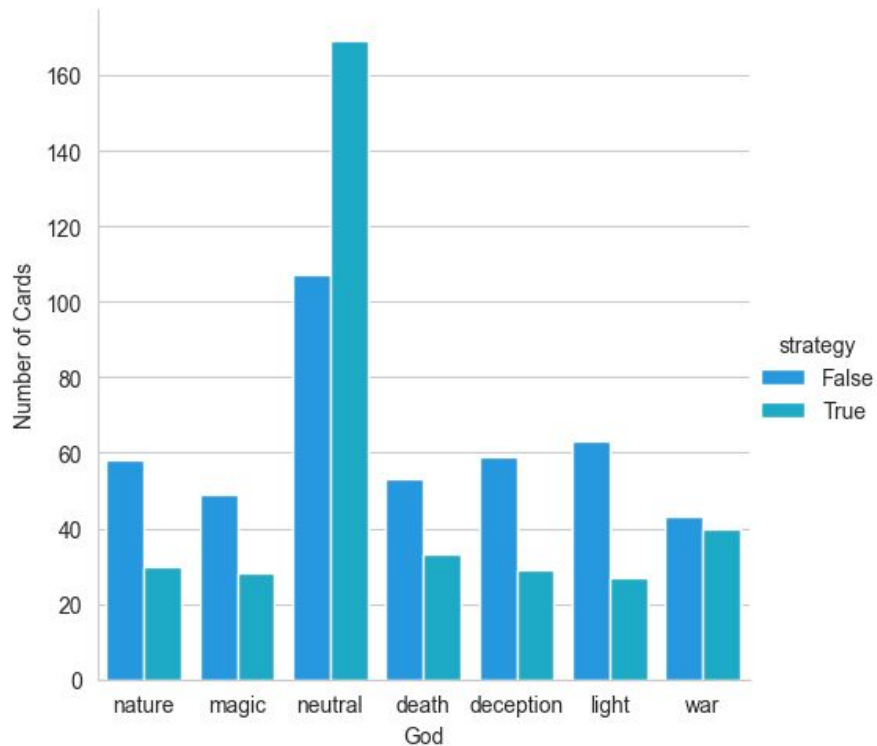
Exploratory Data Analysis

Analysis



Exploratory Data Analysis

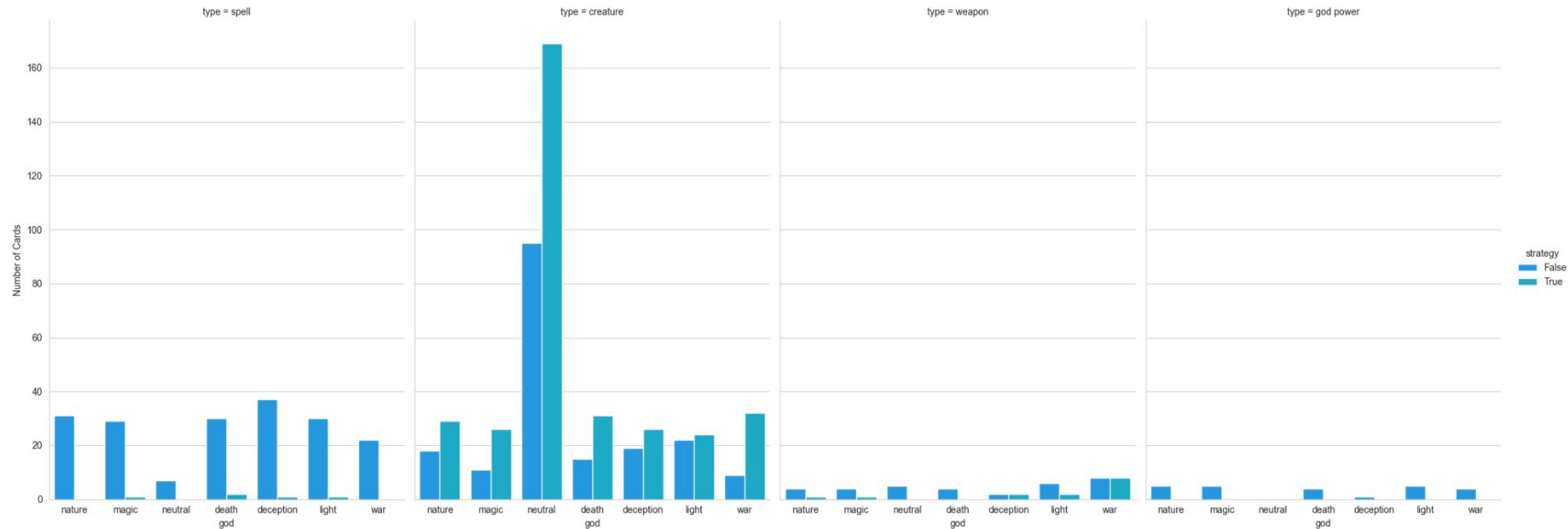
Analysis



Late = True
Early = False

Exploratory Data Analysis

Analysis

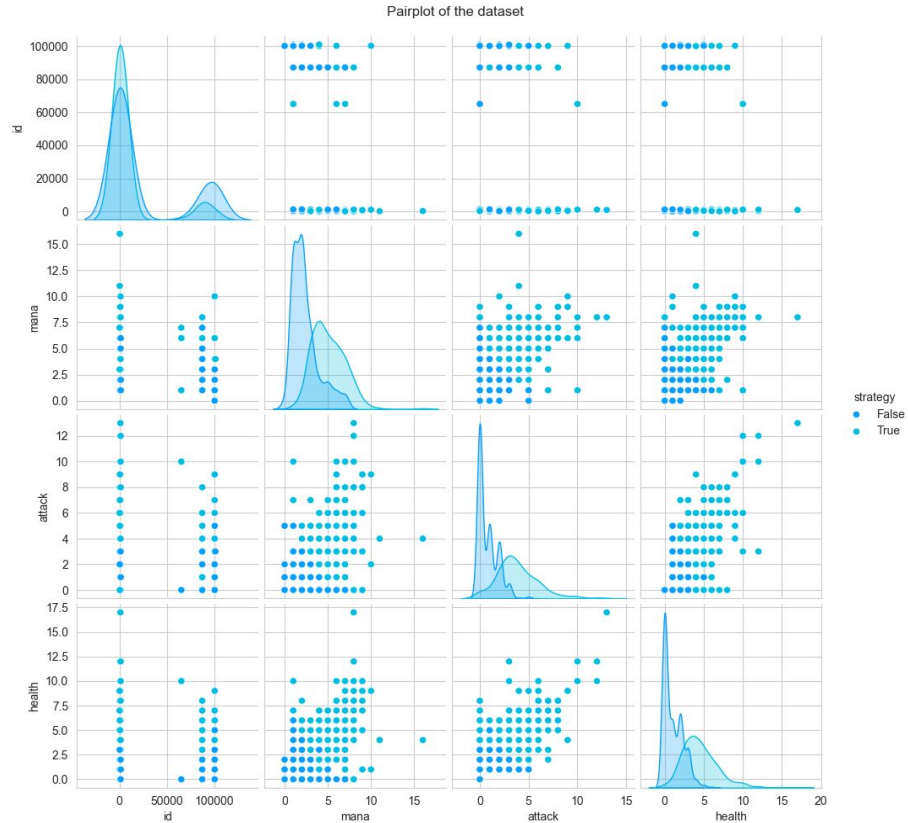


```
strategy type
False creature 189
  spell 186
  weapon 33
  god power 24
True creature 337
  weapon 14
  spell 5
Name: count, dtype: int64
```

Late = True
Early = False

Exploratory Data Analysis

Analysis



Late = True
Early = False

Exploratory Data Analysis

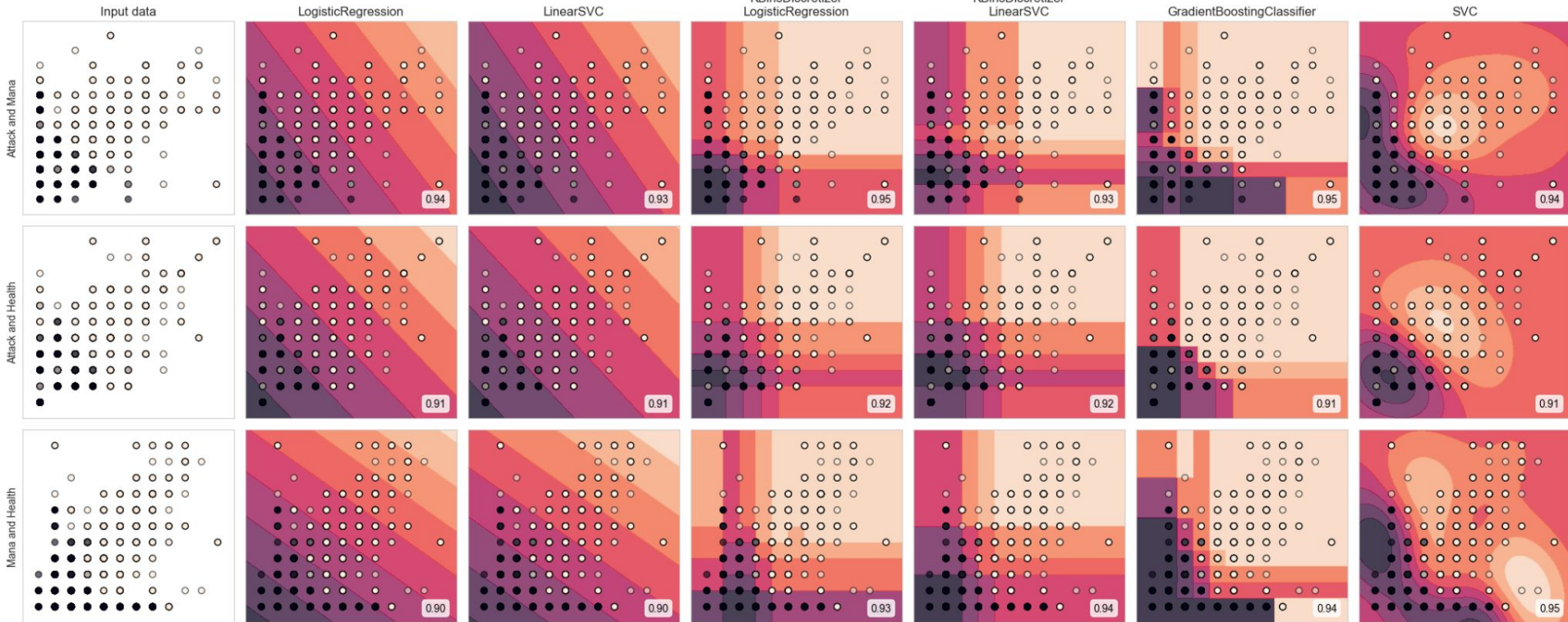
Analysis

Linear

Discretização de Características
com Classificador Linear

Não Linear

Classifiers on feature pairs



Exploratory Data Analysis

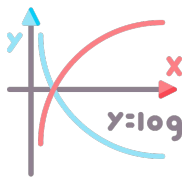
Analysis



Análises

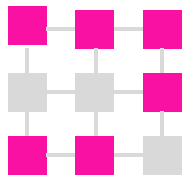
- Forte correlação entre **attack** e **health**, **strategy** e **health**, **attack** e **strategy**, respectivamente.
- As combinações **attack** e **mana** e **health** e **mana** possuem perfis de distribuições **não linear**, pois os melhores discretizadores foram Gradient Boosting e SVC;
- Os campos **health** e **attack** têm como melhor opção o discretizador K-Bins, combinado com LogisticRegression e LinearSVC;
- É possível observar que os campos **attack** e **health** possuem a maior concentração de valores à esquerda;
- A distribuição dos dados é considerada normal para os campos **attack**, **health** e **mana**, mas é possível observar alguns valores discrepantes nos campos.
- A maioria dos tipos de cartas são **creatures**, seguidas de **spells**.
- As cartas **neutral** são as mais comuns na coluna de **gods**, seguidas pelas cartas associadas a diferentes deuses, sem que nenhum deus único domine a distribuição;
- Não há cartas **god power** classificada como **late**.

Feature Engineering



Transformação Logaritmica

Aplicado em attack, mana e health.



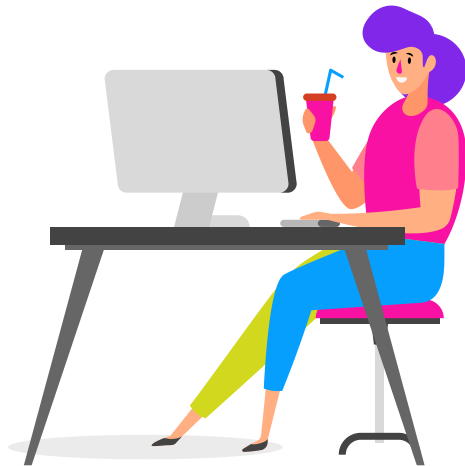
Combinação de Características

Gerando attack_mana, attack_health, mana_health e attack_mana_health.

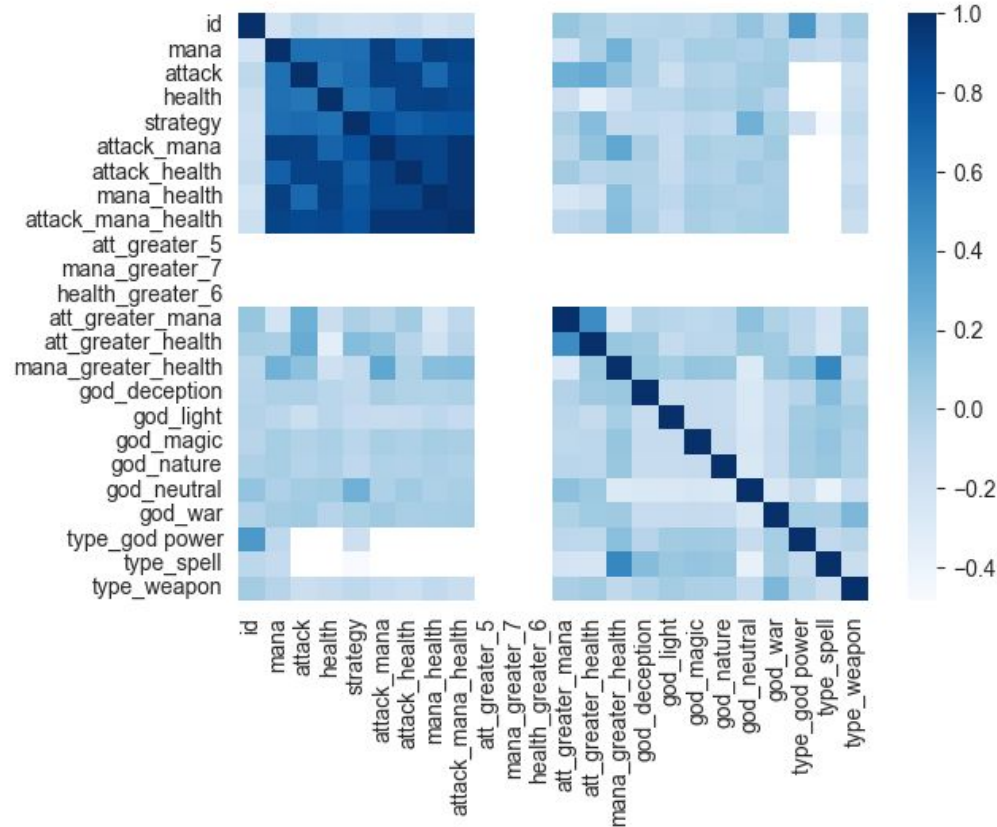


Características Combinadas

Gerando att_greater_5, mana_greater_7, health_greater_6, att_greater_mana, att_freater_health, mana_greater_health

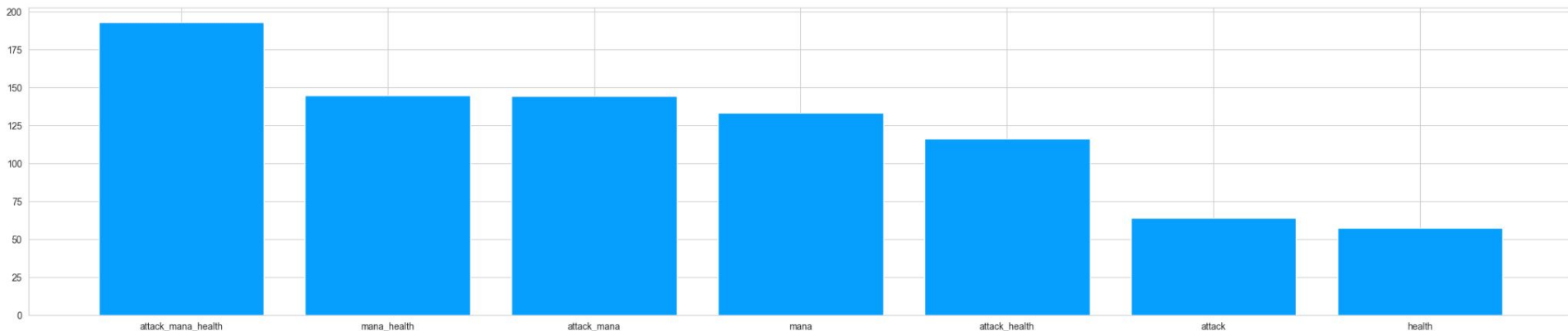


Feature Engineering



Feature Engineering

Select KBest

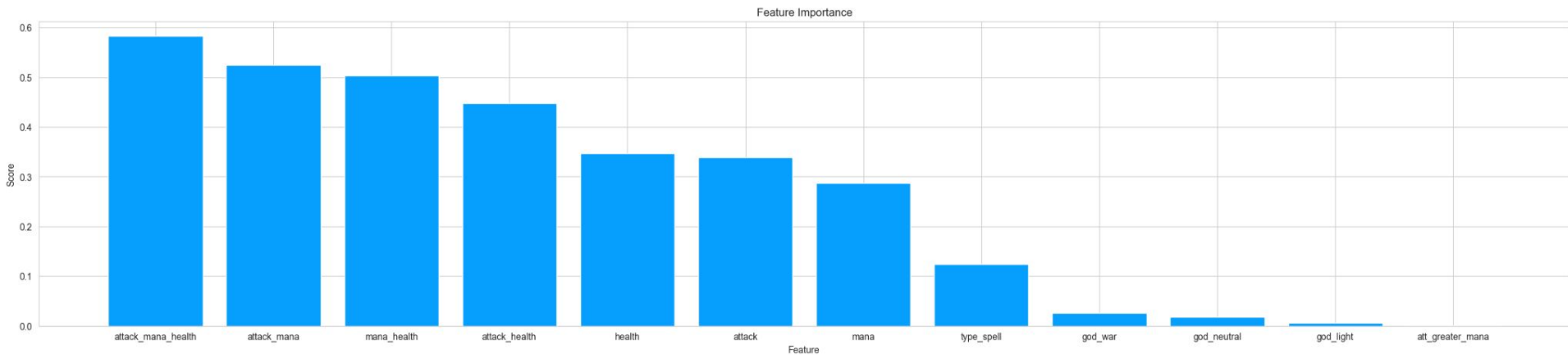


```
('attack_mana_health', 192.90500920601278)
('mana_health', 144.91037613043042)
('attack_mana', 144.2699076506875)
('mana', 133.27924334029575)
('attack_health', 116.05697458589944)
('attack', 64.00187565167143)
('health', 57.523697903414)
('god_neutral', nan)
('type_god power', nan)
('type_spell', nan)
```

Utilizando Teste Qui-Quadrado

Feature Engineering

Mutual Information



attack_mana_health	0.583453
attack_mana	0.525046
mana_health	0.503692
attack_health	0.448270
health	0.347232
attack	0.338541
mana	0.287186
type_spell	0.124505
god_war	0.025890
god_neutral	0.018468
god_light	0.006464
att_greater_mana	0.001243

Model Building

01

Preparação do Dataset de Teste

Adicionando *Feature Engineering* de treino.

03

Divisão e Treinamento

O *dataset* de teste não tem a variável resposta. Utilizou-se o dataset de treinamento.

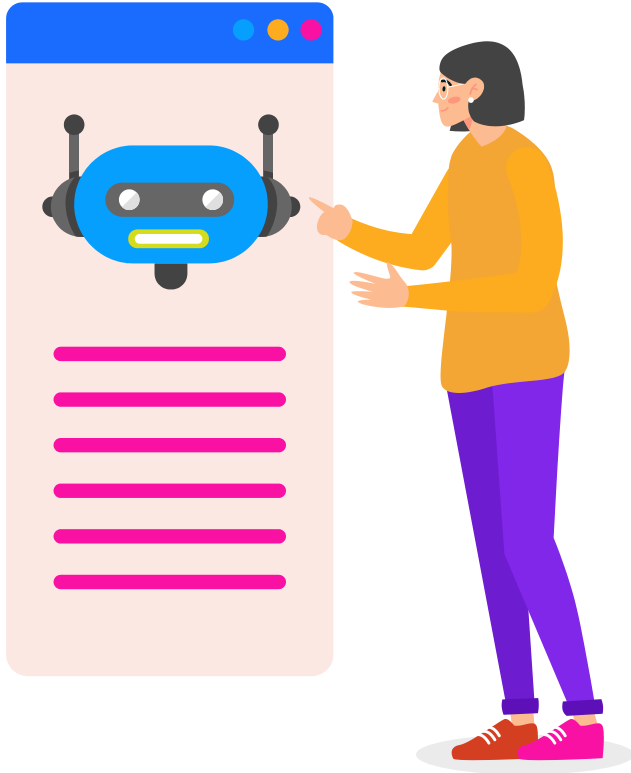
02

Seleção de Modelos

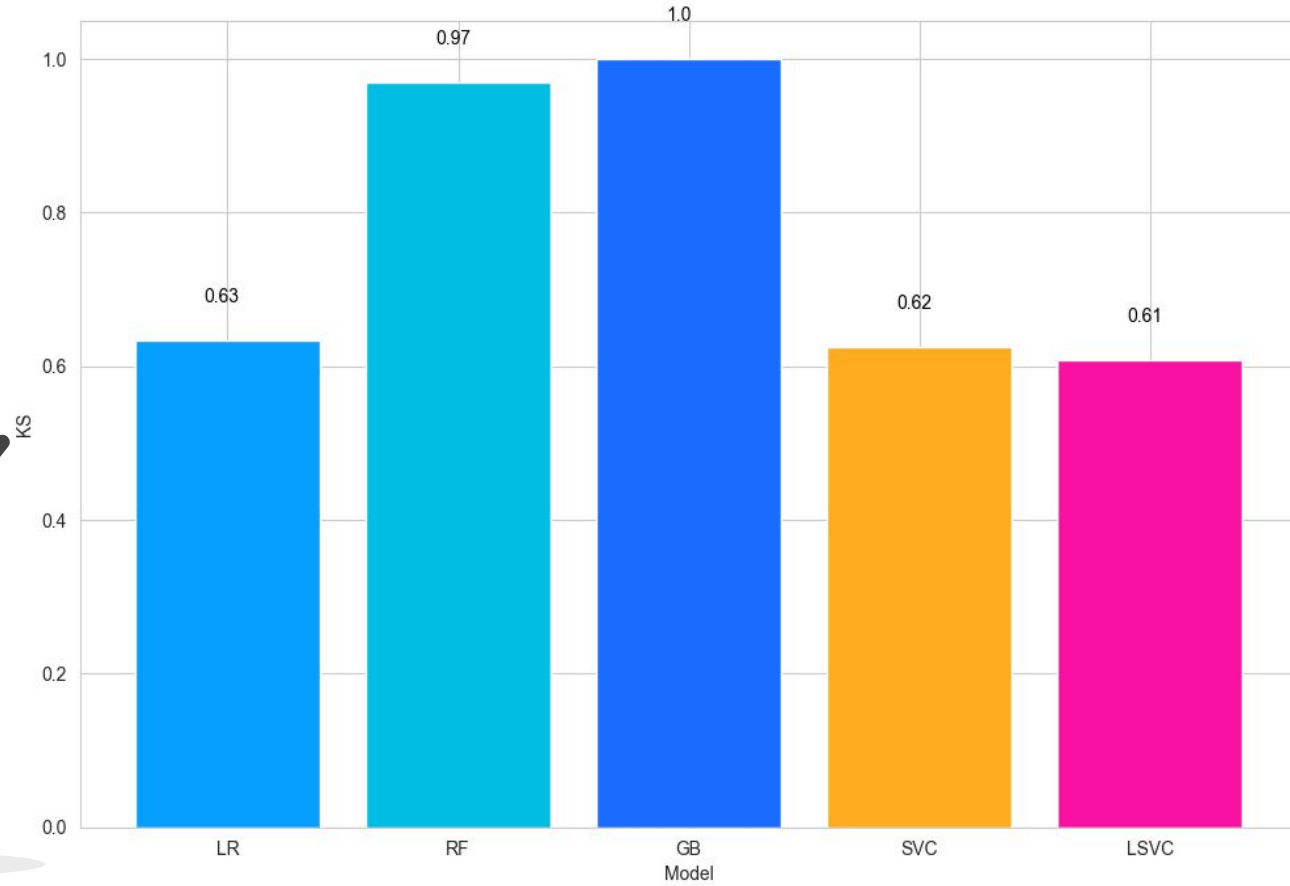
Regressão Logística,
Random Forest,
Gradient Boosting,
Support Vector e
Support Vector Linear

04

Escolha do Melhor Modelo



Models Performance Comparison



Pipeline



aws
Suporte



Curva de
Aprendizado



Financeiro



Apache
Airflow



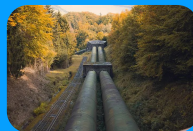
Financeiro



Curva de
Aprendizado



Memória



Criando Pipelines para Produtização de Modelos de Machine Learning
Alvaro Leandro Cavalcante Carneiro



Pipeline

Considerações finais

Será julgado seu poder de síntese, **de fazer mais com menos**, de arquitetar a solução da forma mais simples e robusta possível, de uma forma que seja manutenível, com qualidade e elegância.

Você pode, em caso de dúvidas relacionadas ao desafio, nos encaminhar um email com essas perguntas.

Esperamos que você consiga concluir o desafio em 1 semana, mas se precisar de mais tempo basta nos informar.

Esperamos que se divirta no processo.

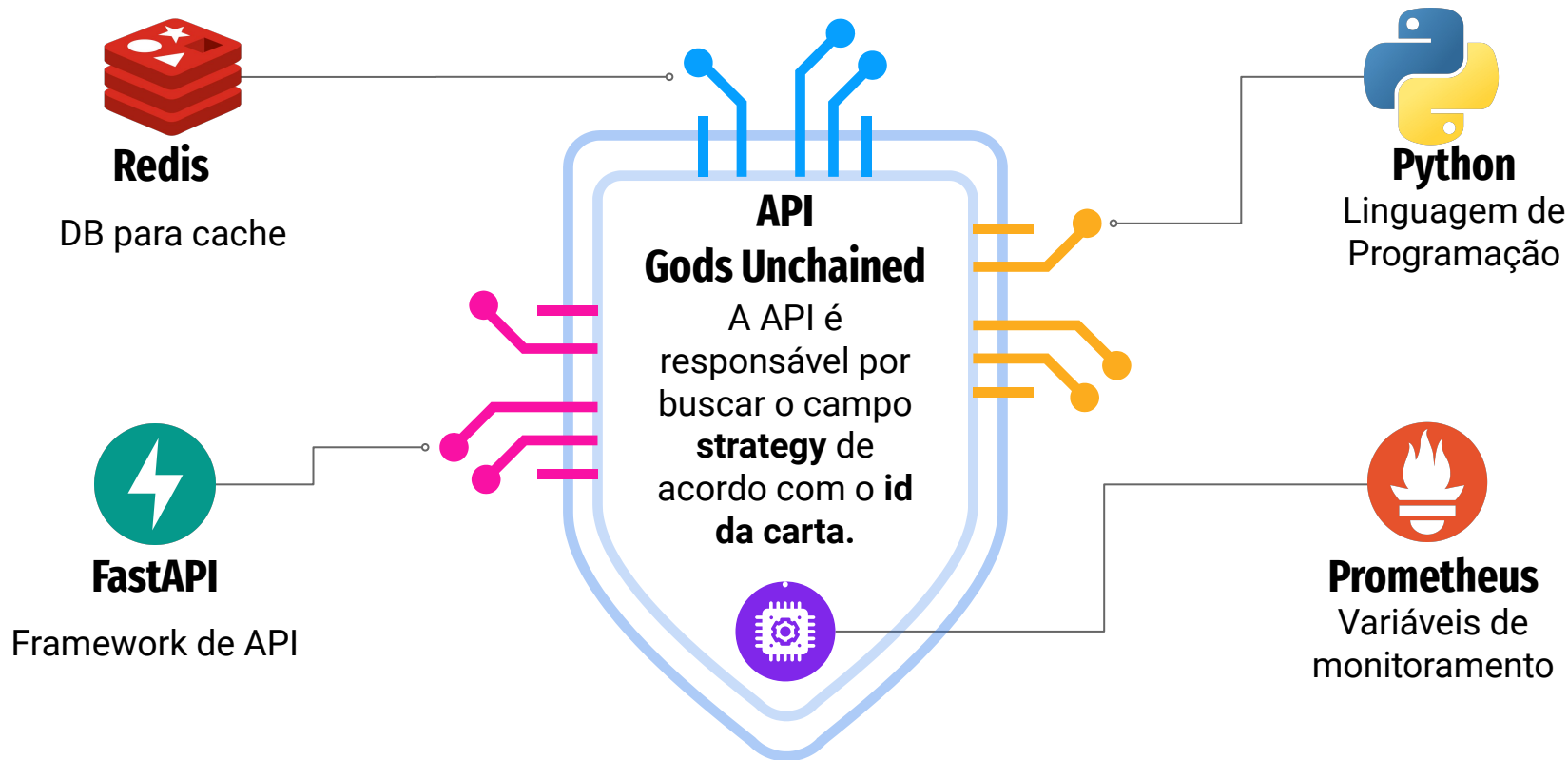
Boa sorte!



GitHub Actions



Application Programming Interface



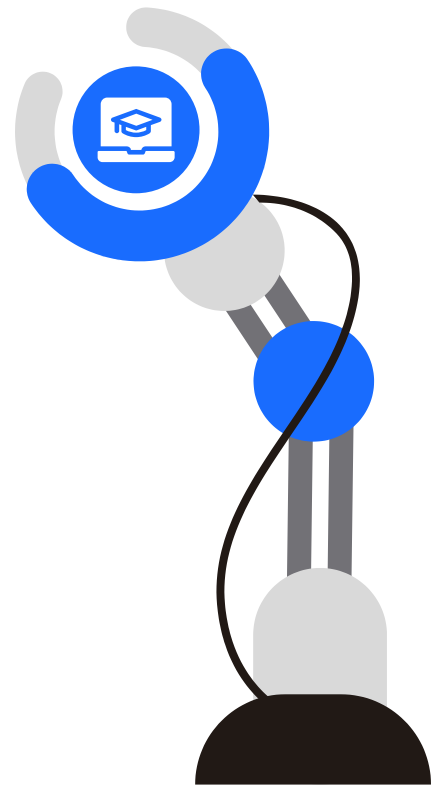
Melhorias e Aprendizados

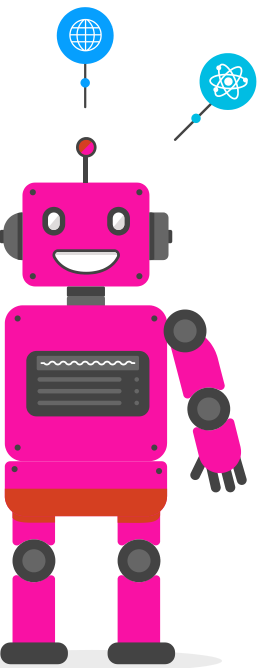
Melhorias

- Dar push nos artefatos dentro da branch main;
- Combinar prometheus com grafana, para monitoramento;
- Adicionar segurança mais robusta;
- Implementação de Feature Store;
- Fazer limpeza do redis a cada deploy de modelo;
- Migrar para AWS Sagemaker ou Airflow.

Aprendizados

- Manifestos para kubernetes;
- Aprender usar Kubeflow;
- Aprender usar DVC;
- Aprender usar Deepchecks.





Obrigado!

Dúvidas?



<https://lucasbrito.com.br>
lucasbsilva29@gmail.com

