

Classificação de partidas de League of Legends usando KNN e Árvore de Decisão

Alunos: Guilherme Alves Carvalho e Heitor Freitas Ferreira

Tarefa de mineração escolhida

Classificação dos dados das partidas competitivas de League Of Legends para descobrir se um time irá ganhar a partida dadas informações da partida

Dados disponíveis no [kaggle](#) e no [github](#)

Base de dados

- Arquivo csv contendo informações dos times e dos jogadores em todas partidas das principais ligas do mundo como LCS, LEC, LCK, LPL, PCS, CBLol e mais
- Todos os dados foram agregados e divulgados por Tim Sevenhuysen da OraclesElixir.com.

Pré processamento

- Base originalmente continha 123 atributos
 - Atributos de players
 - Atributos dos times

'gameid',	'ban3',	'ckpm',	'barons',	'wardsplaced',	'cspm',	'opp_goldat15',
'datacompleteness',	'ban4',	'firstdragon',	'opp_barons',	'wpm',	'goldat10',	'opp_xpat15',
'url',	'ban5',	'dragons',	'firsttower',	'wardskilled',	'xpat10',	'opp_csat15',
'league',	'gamelength',	'opp_dragons',	'towers',	'wcpm',	'csat10',	'golddiffat15',
'year',	'result',	'elementaldrakes',	'opp_towers',	'controlwardsbought',	'opp_goldat10',	'xpdiffat15',
'split',	'kills',	'opp_elementaldrakes',	'firstmidtower',	'visionscore',	'opp_xpat10',	'csdiffat15',
'playoffs',	'deaths',	'infernals',	'firsttothreetowers',	'vspm',	'opp_csat10',	'killsat15',
'date',	'assists',	'mountains',	'turretplates',	'totalgold',	'golddiffat10',	'assistsat15',
'game',	'teamkills',	'clouds',	'opp_turretplates',	'earnedgold',	'xpdiffat10',	'deathsat15',
'patch',	'teamdeaths',	'oceans',	'inhibitors',	'earned_gpm',	'csdiffat10',	'opp_killsat15',
'participantid',	'doublekills',	'chemtechs',	'opp_inhibitors',	'earnedgoldshare',	'killsat10',	'opp_assistsat15',
'side',	'triplekills',	'hextechs',	'damagetochampions',	'goldspent',	'assistsat10',	'opp_deathsat15',
'position',	'quadrakills',	'dragons (type unknown)',	'dpm',	'gspd',	'deathsat10',	'ban2',
'playername',	'pentakills',	'elders',	'damageshare',	'total_cs',	'opp_killsat10',	'firstbaron',
'playerid',	'firstblood',	'opp_elders',	'damagetakenperminute',	'minionkills',	'opp_assistsat10',	
'teamname',	'firstbloodkill',	'firstherald',	'damagemitigatedperminute',	'monsterkills',	'opp_deathsat10',	
'teamid',	'firstbloodassist',	'heralds',		'monsterkillsownjungle',	'goldat15',	
'champion',	'firstbloodvictim',	'opp_heralds',		'monsterkillsenemyjungle',	'xpat15',	
'ban1',	'team kpm',				'csat15',	

Pré processamento

- Base originalmente continha 123 atributos
 - Atributos de players
 - Atributos dos times
- Dados temporais divididos pela duração do jogo
 - totalgold, minionskills, monsterkills, constrolwardsbought
- Todas colunas normalizadas, exceto dados categóricos
 - 'league','game','side','teamname','ban1','ban2','ban3','ban4','ban5','result','firstblood','firstdragon','firsttherald','firstbaron','firsttower','firstmidtower','firstthreetowers'

Pré processamento

- Seleccionadas as linhas das seguintes ligas
 - 'LCK', 'SL', 'LCK CL', 'NLC', 'PRM', 'VCS', 'LMF', 'LCS', 'PCS',
'LFL','CBLLOL', 'LEC', 'LAS', 'TCL', 'LJL', 'LCO', 'LPLLOL', 'NEXO', 'LLA',
'EBL', 'GL', 'PGN', 'WCS','MSI', 'CDF','LCL'
- Seleccionadas linhas com todos os dados
- Removidas linhas que diziam respeito aos players
- Seleccionado apenas partidas de 2022

Colunas com maior correlação com 'result'

golddiffat15	0.540542
firstthreetowers	0.555000
monsterkills_per_minute	0.565751
elementaldrakes	0.586517
dragons	0.587794
doublekills	0.594394
barons	0.628934
firstbaron	0.663992
team kpm	0.681801
assists	0.682777
teamkills	0.686663
kills	0.686663
inhibitors	0.749887
totalgold_per_minute	0.826226
towers	0.885757

Foram usadas como entrada para os modelos de classificação os 15 elementos com a maior correlação

Ao final do pré-processamento, saímos de um dataset com 123 atributos e 149232 objetos, para um com 15 atributos e 12000 objetos

Métodos utilizados

KNN

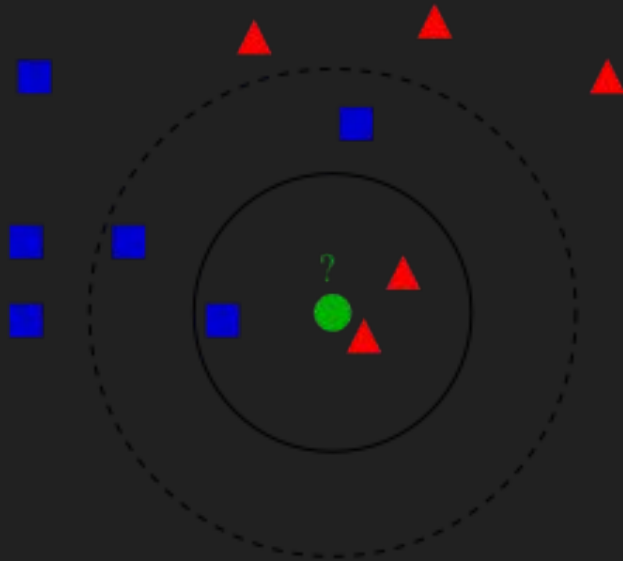
- Aprendizado supervisionado
- Calcula as distâncias de um objeto para todos outros
- O objeto é classificado de acordo com a classificação dos vizinhos
- Muito sensível à ruídos
- Dependente de uma boa escolha da quantidade de vizinhos (k)
 - k grande -> Reduz influência de ruído
 - k pequeno -> Separação melhor entre as classes

Árvore de decisão (ID3)

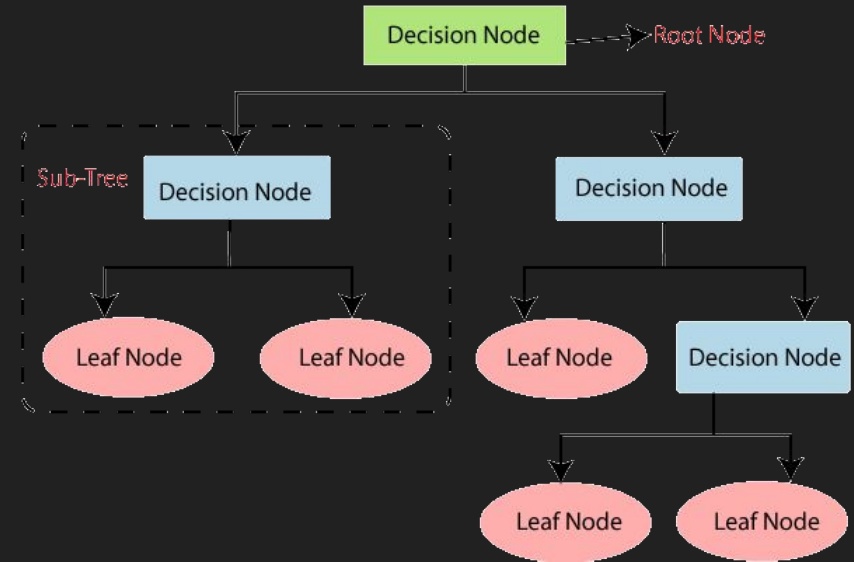
- Aprendizado supervisionado
- Começa com todos os objetos em um nó raiz
- Divide o nó raiz usando o atributo que maximiza o ganho de entropia considerando o nó raiz e seus filhos
- Processo repetido recursivamente até obter apenas elementos de uma classe no nó
- Pode gerar árvore mais complexas que o necessário que não generaliza bem (overfitting)

Métodos utilizados

KNN



Árvore de decisão (ID3)



Classificação usando KNN

- Dataset dividido aleatoriamente em 90% para treino e 10% para teste
- Usada distância de Manhattan
- Testado com $k = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]$
- Para cada teste foi montada a matriz de confusão
- Para cada teste foram calculados os valores de Sensitividade e Especificidade

Valores para diferentes Ks

Para **K = 1**:

Valor real	0	1
------------	---	---

Valor predito		
---------------	--	--

0	563	22
---	-----	----

1	25	590
---	----	-----

Sensitividade = 96.41%

Especificidade = 95.75%

Para **K = 3**:

Valor real	0	1
------------	---	---

Valor predito		
---------------	--	--

0	563	12
---	-----	----

1	25	600
---	----	-----

Sensitividade = 98.04%

Especificidade = 95.75%

Para **K = 5**:

Valor real	0	1
------------	---	---

Valor predito		
---------------	--	--

0	564	9
---	-----	---

1	24	603
---	----	-----

Sensitividade = 98.53%

Especificidade = 95.92%

Para **K = 7**:

Valor real	0	1
------------	---	---

Valor predito		
---------------	--	--

0	557	5
---	-----	---

1	31	607
---	----	-----

Sensitividade = 99.18%

Especificidade = 94.73%

Para **K = 9**:

Valor real	0	1
------------	---	---

Valor predito		
---------------	--	--

0	563	8
---	-----	---

1	25	604
---	----	-----

Sensitividade = 98.69%

Especificidade = 95.75%

Para **K = 11**:

Valor real	0	1
------------	---	---

Valor predito		
---------------	--	--

0	562	8
---	-----	---

1	26	604
---	----	-----

Sensitividade = 98.69%

Especificidade = 95.58%

Valores para diferentes Ks

Para **K = 13**:

Valor real 0 1

Valor predito

0 552 4

1 36 608

Sensitividade = 99.35%

Especificidade = 93.88%

Para **K = 17**:

Valor real 0 1

Valor predito

0 553 4

1 35 608

Sensitividade = 99.35%

Especificidade = 94.05%

Para **K = 15**:

Valor real 0 1

Valor predito

0 555 4

1 33 608

Sensitividade = 99.35%

Especificidade = 94.39%

Para **K = 19**:

Valor real 0 1

Valor predito

0 554 4

1 34 608

Sensitividade = 99.35%

Especificidade = 94.22%

Classificação usando Árvore de Decisão (ID3)

Valor real 0 1

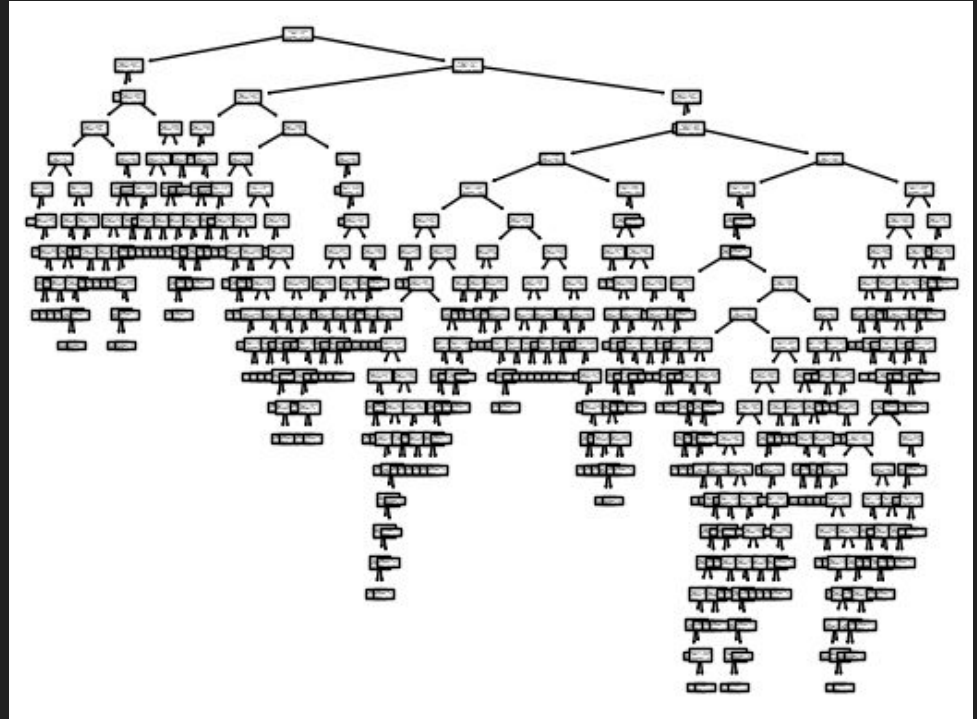
Valor predito

0 588 20

1 17 563

Sensitividade = 96.57%

Especificidade = 97.19%



Conclusão

Ambos os modelos tiveram resultados satisfatórios com relação à classificação da partida em vitória ou derrota usando os top 15 atributos mais relacionados com a variável alvo (resultado da partida), futuramente em outra pesquisa o grupo pretende usar a mesma base para agrupar os jogadores conforme seus dados individuais ao longo de todas as ligas e realizar a classificação para prever o resultado com os dados coletados no meio da partida (15 minutos de jogo).