
Ciência de Dados
no Esporte
Prevendo Vencedores de jogos da NBA

- Trabalho de Conclusão de Curso
- PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
- Curso: Pós-graduação Lato Sensu em Ciência de Dados e Big Data
- Aluno: Geraldo Rodrigues Novais Junior

2020

Apresentação

- Objetivo;
- Coleta dos Dados;
- Ferramentas;
- Processamento / Tratamento dos Dados;
- Análise e Exploração dos Dados;
- Tratamento Complementar dos Dados;
- Criação de Modelos de Machine Learning;
- Resultados.

Objetivo

- Confirmar se é possível, além da aleatoriedade ou do mero palpite, indicar o favorito a vencer um jogo da NBA, com base em números e estatísticas coletadas durante uma temporada específica.

Não há pretensão de se construir modelos com alta capacidade de precisão (afinal se isso fosse possível o torneio já teria perdido prestígio por ser altamente previsível). O nível de competitividade na NBA é alto e até mesmo em seus regulamentos existem dispositivos que visam manter o equilíbrio entre as equipes ao limitar valores para contratações, entre outros. A ideia é apontar o favoritismo de uma das equipes com base em dados, conceitos estatísticos e aprendizado de máquina.

Coleta dos Dados

- Origem

Os conjuntos analisados foram retirados do site de competições Kaggle em 23 de junho de 2020. Foram criados por Nathan Lauga e Ionas Kelepouris que mineraram os dados dos sites da NBA e BasketBall Reference, respectivamente.

- Conjuntos

- Conjunto games.csv

Conjunto de dados com todos os jogos da NBA da temporada 2003/2004 a fevereiro de 2020.

- Conjunto teams.csv

Possui os dados das equipes. Se trata de um complemento ao contido no arquivo "games.csv“.

- Conjunto nba.games.stats.csv

Conjunto com jogos da NBA da temporada 2014 a 2018. Neste novo conjunto temos atributos que não existem no primeiro arquivo "games.csv".

Ferramentas

Processamento / Tratamento de Dados

Conjunto games.csv

- 23195 Registros;
- 21 Atributos;
- 99 Missing (excluídos)
- Sem dados duplicados

+

Conjunto teams.csv

- 30 Registros;
- 14 Atributos;
- 4 Missing (excluídos)
- Sem dados duplicados

Base Histórica

(Treinamento)

	season	home	away	fg_pct_home	ft_pct_home	fg3_pct_home	ast_home	reb_home	fg_pct_away	ft_pct_away	fg3_pct_away	ast_away	reb_away	home_team_wins
0	2003	MIL	POR	0.494	0.762	0.500	31.0	35.0	0.483	1.000	0.250	24.0	42.0	0
1	2003	CHI	GSW	0.390	0.576	0.444	22.0	60.0	0.273	0.633	0.308	18.0	56.0	1
2	2003	BKN	MIA	0.382	0.755	0.188	16.0	48.0	0.380	0.893	0.467	19.0	37.0	1
3	2003	MEM	NOP	0.493	0.704	0.444	24.0	46.0	0.411	0.750	0.263	20.0	49.0	1
4	2003	LAC	NYK	0.413	0.828	0.353	16.0	39.0	0.425	0.909	0.182	18.0	39.0	1

Processamento / Tratamento de Dados

Conjunto nba.games.stats.csv

- 9840 Registros;
- 41 Atributos;
- Sem dados missing
- Sem dados duplicados

Base de Previsões

(Testes)

	season	home	away	fg_pct_home	ft_pct_home	fg3_pct_home	ast_home	reb_home	fg_pct_away	ft_pct_away	fg3_pct_away	ast_away	reb_away	home_team_wins
0	2017	IND	POR	0.520	0.781	0.265	29.0	47.0	0.489	0.786	0.583	22.0	57.0	0
1	2017	CLE	ORL	0.458	0.840	0.227	19.0	50.0	0.506	0.867	0.545	30.0	40.0	0
2	2017	MEM	GSW	0.424	0.571	0.310	19.0	45.0	0.516	0.800	0.450	29.0	49.0	1
3	2017	MIL	POR	0.463	0.833	0.314	23.0	30.0	0.486	0.778	0.535	20.5	54.0	1
4	2017	TOR	PHI	0.470	0.929	0.448	26.0	49.0	0.462	0.737	0.429	25.0	48.0	1

Analise Exploratória dos Dados

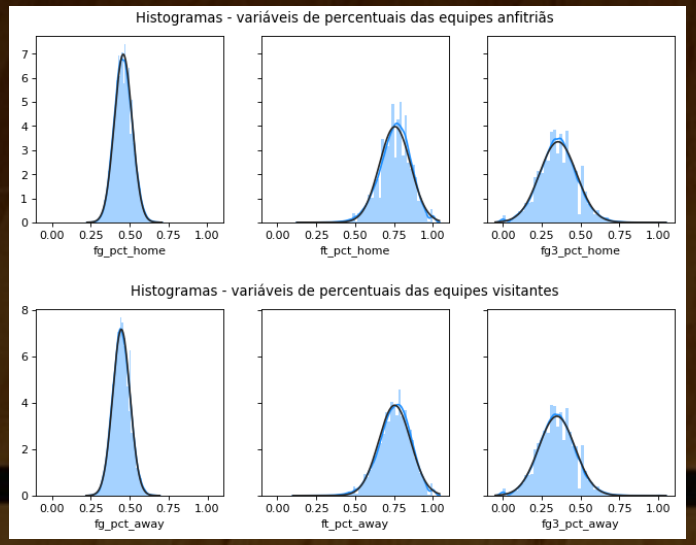
Descrição estatística dos atributos da Base Histórica.

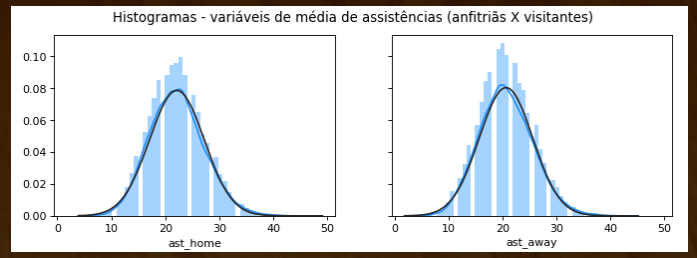
	fg_pct_home	ft_pct_home	fg3_pct_home	ast_home	reb_home	fg_pct_away	ft_pct_away	fg3_pct_away	ast_away	reb_away	home_team_wins
count	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000	19371.000000
mean	0.458978	0.756240	0.354782	22.142688	42.784059	0.446894	0.754068	0.347650	20.657891	41.457746	0.598627
std	0.057131	0.100307	0.119115	5.073358	6.521216	0.055781	0.102608	0.116857	4.961087	6.406019	0.490189
min	0.250000	0.167000	0.000000	6.000000	15.000000	0.244000	0.143000	0.000000	4.000000	19.000000	0.000000
25%	0.420000	0.692000	0.278000	19.000000	38.000000	0.409000	0.690000	0.273000	17.000000	37.000000	0.000000
50%	0.458000	0.762000	0.353000	22.000000	43.000000	0.446000	0.760000	0.348000	20.000000	41.000000	1.000000
75%	0.500000	0.824000	0.429000	25.000000	47.000000	0.484000	0.826000	0.423000	24.000000	46.000000	1.000000
max	0.684000	1.000000	1.000000	47.000000	72.000000	0.670000	1.000000	1.000000	43.000000	81.000000	1.000000

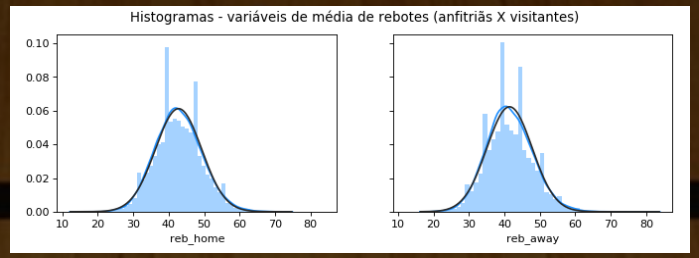
Descrição estatística dos atributos da Base de Previsões.

	fg_pct_home	ft_pct_home	fg3_pct_home	ast_home	reb_home	fg_pct_away	ft_pct_away	fg3_pct_away	ast_away	reb_away	home_team_wins
count	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000
mean	0.462912	0.768085	0.359173	23.145767	43.976697	0.453438	0.767769	0.363091	22.153898	42.956412	0.579212
std	0.022888	0.039246	0.027710	2.792130	2.869937	0.020902	0.040108	0.028752	2.312572	2.636949	0.493893
min	0.341000	0.524000	0.227000	13.500000	30.000000	0.354000	0.625000	0.083000	15.000000	34.000000	0.000000
25%	0.450000	0.746000	0.345000	21.300000	42.400000	0.441000	0.746000	0.353000	20.900000	41.100000	0.000000
50%	0.463000	0.770000	0.360000	22.900000	43.900000	0.450000	0.768000	0.363000	21.900000	43.100000	1.000000
75%	0.477000	0.787000	0.375000	24.500000	45.300000	0.465000	0.796000	0.375000	23.000000	44.800000	1.000000
max	0.548000	1.000000	0.533000	34.000000	63.000000	0.550000	1.000000	0.583000	31.200000	59.000000	1.000000

Histogramas – Base Histórica







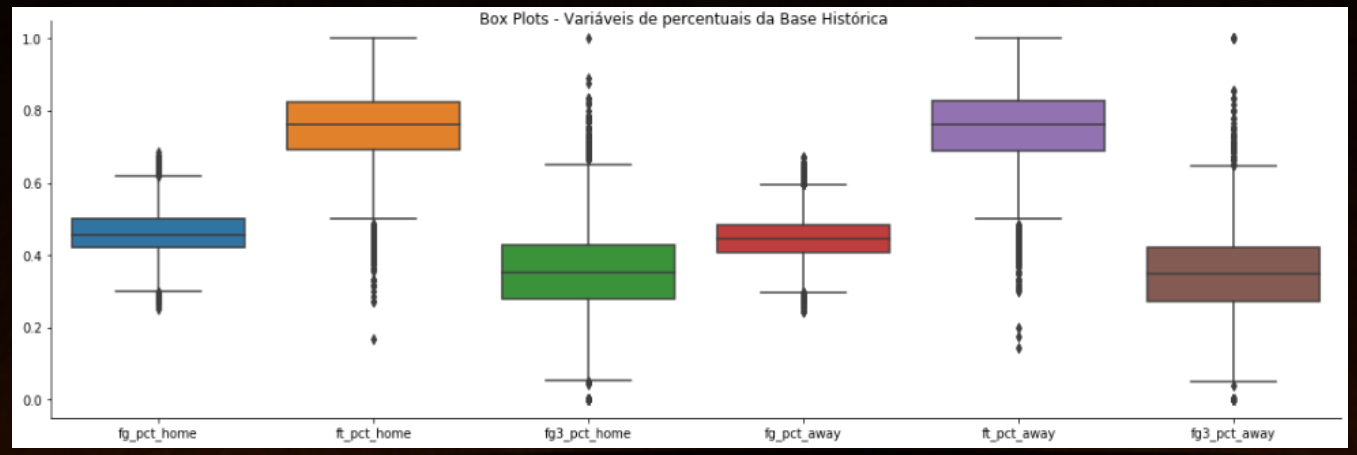
Descrição dos conjuntos

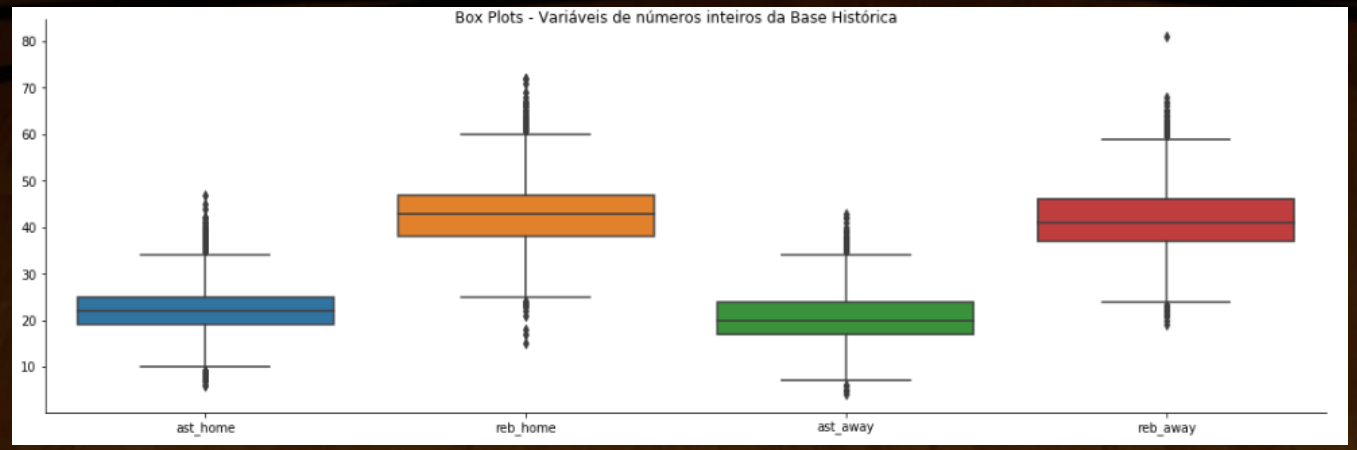
- Conhecer os atributos
- Estatísticas próximas entre os conjuntos
- Diferentes escalas

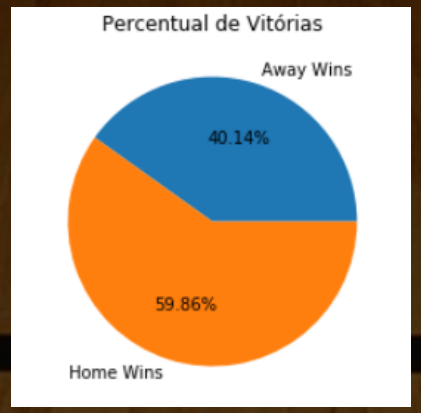
Distribuição

- Distribuição Normal
- Curvas das estatísticas de equipes anfitriãs e visitantes são semelhantes.
- Mesma tendência na Base de Previsões

Box Plots







Analise Exploratória dos Dados

Avaliação de “outliers”

- Atributos com pontos outliers
- Pontos entre anfitriãs e visitantes são semelhantes.
- Não aparentam erro de cadastro ou ingestão

Balanceamento das Classes

- Desbalanceamento entre classes
- Maior volume de vitórias de anfitrãos

Tratamento Complementar dos Dados

“Rebalanceamento” das Classes

- Técnica de “reamostragem”
- Under Sampling
- Elimina aleatoriamente entradas da classe com maior número de ocorrências

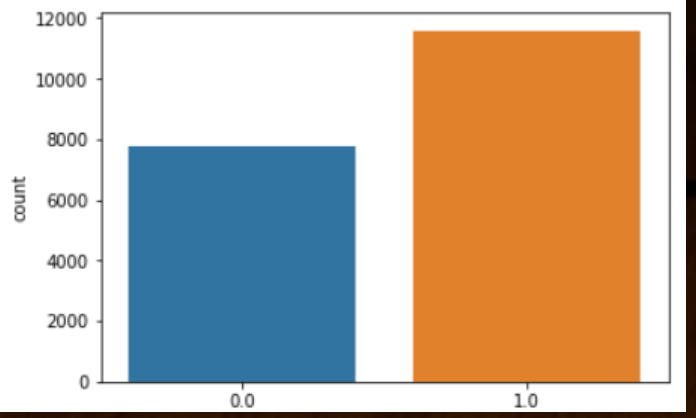
“Padronização”

- Redimensionamento dos dados
- Média zero e desvio padrão igual a um
- Melhora a condição numérica dos problemas de otimização

Distribuição das classes na Base Histórica

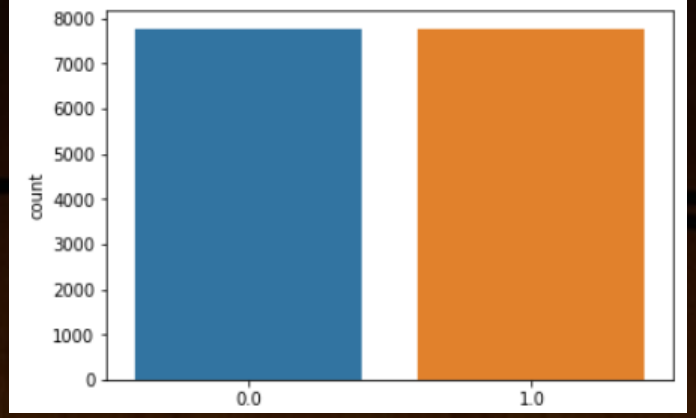
Antes da reamostragem

```
Quantidade por classe:
1.0    11596
0.0     7775
dtype: int64
```



Depois da reamostragem

```
Quantidade por classe:
1.0    7775
0.0    7775
dtype: int64
```



Comparação de amostras dos dados originais com dados padronizados

```
Dados Originais:

 [[ 0.494  0.762  0.5   31.    35.     0.483  1.     0.25  24.    42.   ]
 [ 0.374  0.828  0.28  23.    45.     0.5    0.563  0.4   19.    44.   ]
 [ 0.462  0.769  0.2   20.    39.     0.482  0.806  0.429 22.    50.   ]]

Dados Padronizados:

 [[ 0.70172996  0.07705628  1.29125152  1.82224484 -1.14929053  0.56174413
   2.3875595  -0.88973464  0.60891105  0.04094739]
 [-1.40383906  0.73047162 -0.56615889  0.23566567  0.38522839  0.86577163
  -1.88586573  0.38770616 -0.39131785  0.35406859]
 [ 0.14024489  0.14635791 -1.24158086 -0.35930151 -0.53548296  0.54386016
   0.49043251  0.63467804  0.20881949  1.2934322 ]]
```

Criação de Modelos de Machine Learning

- Logist Regression
- Random Forest

Resultado da aplicação de regressão logística em dados futuros
(split 80/treino e 20/teste da Base Histórica)

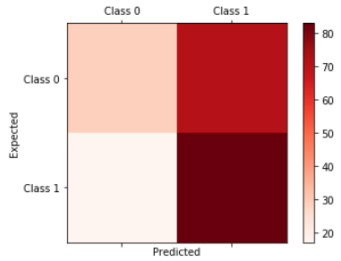
```
Acurácia Modelo de Teste: 83.948%
```

Resultado da aplicação dos modelos na Base de Previsões

```
Regressão Logística
METRICAS MODELO 1.0 - Dados Originais:

Confusion Matrix:
 [[144 358]
 [117 574]]

Percentage Rating (by class):
```

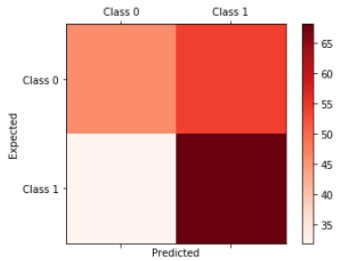



```
Accuracy : 60.18%
Sensitivity : 28.69%
Specificity : 83.07%
```

```
Regressão Logística
METRICAS MODELO 1.1 - Dados Balanceados:

Confusion Matrix:
 [[231 271]
 [220 471]]

Percentage Rating (by class):
```

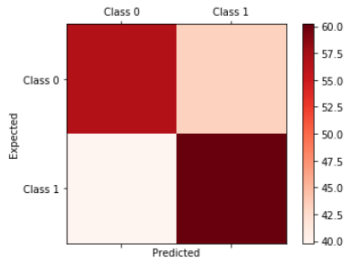



```
Accuracy : 58.84%
Sensitivity : 46.02%
Specificity : 68.16%
```

```
Regressão Logística
METRICAS MODELO 1.2 - Dados Padronizados:

Confusion Matrix:
 [[285 217]
 [275 416]]

Percentage Rating (by class):
```

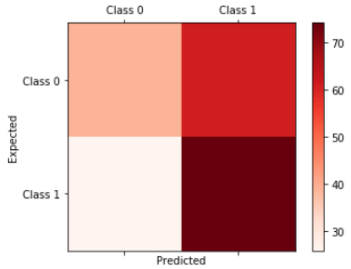



```
Accuracy : 58.76%
Sensitivity : 56.77%
Specificity : 60.20%
```

```
Random Forest
METRICAS MODELO 2.0 - Dados Originais:

Confusion Matrix:
 [[195 307]
 [178 513]]

Percentage Rating (by class):
```

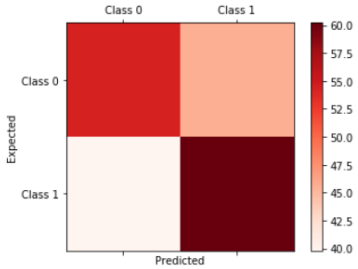



```
Accuracy : 59.35%
Sensitivity : 38.84%
Specificity : 74.24%
```

```
Random Forest
METRICAS MODELO 2.1 - Dados Balanceados:

Confusion Matrix:
 [[273 229]
 [275 416]]

Percentage Rating (by class):
```

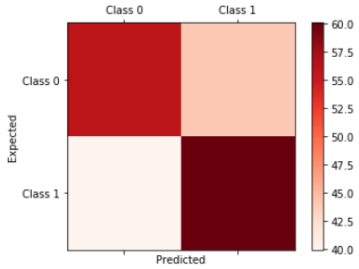



```
Accuracy : 57.75%
Sensitivity : 54.38%
Specificity : 60.20%
```

```
Random Forest
METRICAS MODELO 2.2 - Dados Padronizados:

Confusion Matrix:
 [[281 221]
 [276 415]]

Percentage Rating (by class):
```



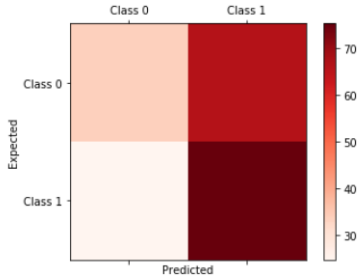

```
Accuracy : 58.34%
Sensitivity : 55.98%
Specificity : 60.06%
```

Resultado da aplicação dos modelos na Base de Previsões

SVM
METRICAS MODELO 3.0 - Dados Originais:

Confusion Matrix:
[[169 333]
[171 520]]

Percentage Rating (by class):

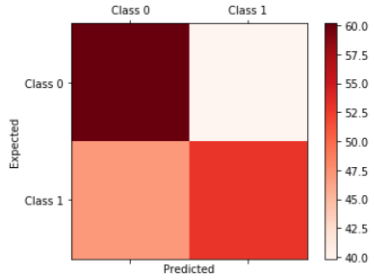


Accuracy : 57.75%
Sensitivity : 33.67%
Specificity : 75.25%

SVM
METRICAS MODELO 3.1 - Dados Balanceados:

Confusion Matrix:
[[302 200]
[325 366]]

Percentage Rating (by class):

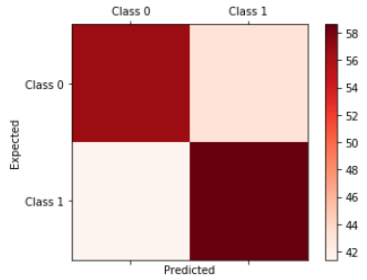


Accuracy : 55.99%
Sensitivity : 60.16%
Specificity : 52.97%

SVM
METRICAS MODELO 3.2 - Dados Padronizados:

Confusion Matrix:
[[285 217]
[286 405]]

Percentage Rating (by class):

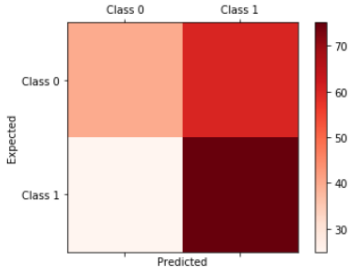


Accuracy : 57.84%
Sensitivity : 56.77%
Specificity : 58.61%

XGBoost
METRICAS MODELO 4.0 - Dados Originais:

Confusion Matrix:
[[200 302]
[172 519]]

Percentage Rating (by class):

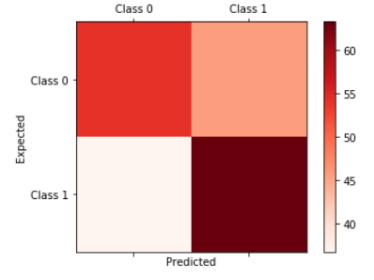


Accuracy : 60.27%
Sensitivity : 39.84%
Specificity : 75.11%

XGBoost
METRICAS MODELO 4.1 - Dados Balanceados:

Confusion Matrix:
[[272 230]
[254 437]]

Percentage Rating (by class):

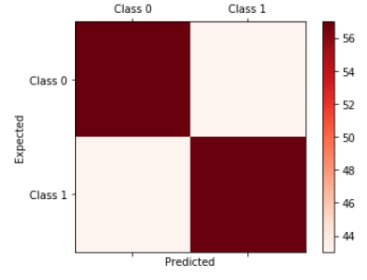


Accuracy : 59.43%
Sensitivity : 54.18%
Specificity : 63.24%

XGBoost
METRICAS MODELO 4.2 - Dados Padronizados:

Confusion Matrix:
[[286 216]
[298 393]]

Percentage Rating (by class):



Accuracy : 56.92%
Sensitivity : 56.97%
Specificity : 56.87%

Criação de Modelos de Machine Learning

- Support Vector Machines
- XGBoost

Criação de Modelos de Machine Learning

“Grid Search Parameter Tuning”

```
# Criando modelo
modeloRL = LogisticRegression(random_state=2403)

# Criando o grid de valores
valores_gridRL = {'penalty': ['l1','l2'],
                  'C': [0.1, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20],
                  'max_iter': [100, 150, 200, 250, 300, 350, 400],
                  'tol': [0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.001, 0.005]}

# Configurando grid
gridRL = GridSearchCV(estimator = modeloRL,
                      param_grid = valores_gridRL,
                      cv = 10,
                      n_jobs = 10,
                      scoring = 'accuracy',
                      verbose = True)

# Treinamento (Buscando melhor configuração de hiperparametros)
gridRL.fit(X_standard, Y_balanced)
```

```
Fitting 10 folds for each of 1372 candidates, totalling 13720 fits
```

```
# Criando modelo
modeloSVM = SVC(random_state=2403)

# Criando o grid de valores
valores_gridSVM = [{'kernel': ['rbf'],
                    'gamma': [1e-3, 1e-4],
                    'C': [0.1, 1, 10, 100, 1000]},
                   {'kernel': ['linear'],
                    'C': [0.1, 1, 10, 100, 1000]}]

# Configurando grid
gridSVM = GridSearchCV(estimator = modeloSVM,
                       param_grid = valores_gridSVM,
                       cv = 10,
                       n_jobs = 10,
                       scoring = 'accuracy',
                       verbose = True)

# Treinamento (Buscando melhor configuração de hiperparametros)
gridSVM.fit(X_standard, Y_balanced)
```

```
Fitting 10 folds for each of 15 candidates, totalling 150 fits
```

```
# Criando modelo
modeloRF = RandomForestClassifier(random_state=2403)

# Criando o grid de valores
valores_gridRF = {'max_features': ['auto', 'sqrt'],
                  'max_depth': [None, 5, 10, 20, 50],
                  'min_samples_split': [2, 5, 10],
                  'min_samples_leaf': [1, 2, 4],
                  'bootstrap': [True, False],
                  'n_estimators': [100, 200, 300]}

# Configurando grid
gridRF = GridSearchCV(estimator = modeloRF,
                      param_grid = valores_gridRF,
                      cv = 10,
                      n_jobs = 10,
                      scoring = 'accuracy',
                      verbose = True)

# Treinamento (Buscando melhor configuração de hiperparametros)
gridRF.fit(X_standard, Y_balanced)
```

```
Fitting 10 folds for each of 540 candidates, totalling 5400 fits
```

```
# Criando modelo
modeloXGB = XGBClassifier(objective= 'binary:logistic', nthread=4, seed=2403)

# Criando o grid de valores
valores_gridXGB = {'max_depth': [3, 4, 5, 6, 8, 10],
                   'n_estimators': [100, 200, 300],
                   'learning_rate': [0.01, 0.05, 0.1, 0.5],
                   'min_child_weight' : [1, 3, 5, 7],
                   'gamma': [0.0, 0.1, 0.2 , 0.3],
                   'colsample_bytree' : [ 0.3, 0.4, 0.5]}

# Configurando grid
gridXGB = GridSearchCV(estimator = modeloXGB,
                       param_grid = valores_gridXGB,
                       cv = 10,
                       n_jobs = 10,
                       scoring = 'accuracy',
                       verbose = True)

# Treinamento (Buscando melhor configuração de hiperparametros)
gridXGB.fit(X_standard, Y_balanced)
```

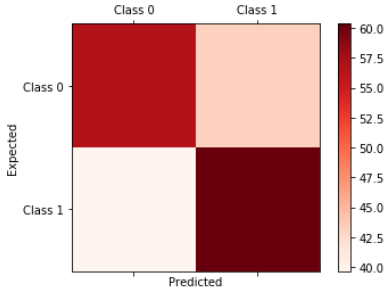
```
Fitting 10 folds for each of 3456 candidates, totalling 34560 fits
```

Resultado da aplicação dos modelos na Base de Previsões

Regressão Logística
METRICAS MODELO 1.3 - Otimizado:

Confusion Matrix:
[[285 217]
[274 417]]

Percentage Rating (by class):

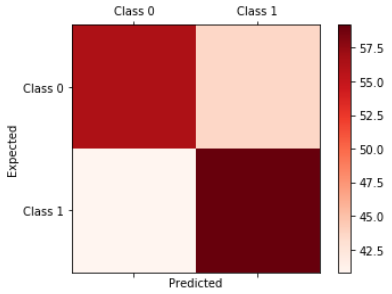


Accuracy : 58.84%
Sensitivity : 56.77%
Specificity : 60.35%

Random Forest
METRICAS MODELO 2.3 - Otimizado:

Confusion Matrix:
[[283 219]
[282 409]]

Percentage Rating (by class):

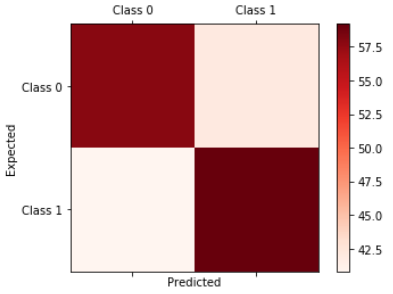


Accuracy : 58.01%
Sensitivity : 56.37%
Specificity : 59.19%

SVM
METRICAS MODELO 3.3 - Otimizado:

Confusion Matrix:
[[291 211]
[282 409]]

Percentage Rating (by class):

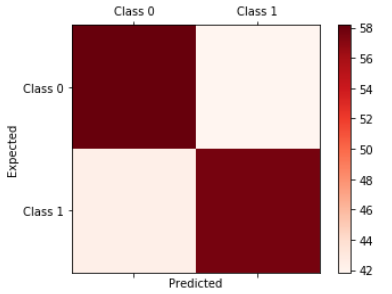


Accuracy : 58.68%
Sensitivity : 57.97%
Specificity : 59.19%

XGBoost
METRICAS MODELO 4.3 - Otimizado:

Confusion Matrix:
[[292 210]
[293 398]]

Percentage Rating (by class):



Accuracy : 57.84%
Sensitivity : 58.17%
Specificity : 57.60%

Criação de Modelos de Machine Learning

Modelos Otimizados

- Logist Regression
- Random Forest
- Support Vector Machines
- XGBoost

Resultados

- Resultados semelhantes
- Sem grandes oscilações
- Opção pelo modelo de Regressão otimizado.

Algoritmo	Metrica	Dados Originais	Dados Balanceados	Dados Padronizados	Dados Padronizados & Modelo Otimizado
Logistic Regression	Accuracy	**60,18%**	**58,84%**	**58,76%**	**58,84%**
	Sensitivity	28,69%	46,02%	56,77%	56,77%
	Specificity	83,07%	68,16%	60,20%	60,35%
Random Forest	Accuracy	**59,35%**	**57,75%**	**58,34%**	**58,01%**
	Sensitivity	38,84%	54,38%	55,98%	56,37%
	Specificity	74,24%	60,20%	60,06%	59,19%
Support Vector Machines	Accuracy	**57,75%**	**55,99%**	**57,84%**	**58,68%**
	Sensitivity	33,67%	60,16%	56,77%	57,97%
	Specificity	75,25%	52,97%	58,61%	59,19%
XGBoost	Accuracy	**60,27%**	**59,43%**	**56,92%**	**57,84%**
	Sensitivity	39,84%	54,18%	56,97%	58,17%
	Specificity	75,11%	63,24%	56,87%	57,60%

Sim, podemos indicar o favorito a vencer um jogo da NBA, além da aleatoriedade ou do mero palpite, com base em números e estatísticas coletadas durante uma temporada específica.

O modelo selecionado, ou mesmo um “blend” dos modelos testados, poderiam ser monetizados ao alimentar um site de sugestão de apostas, de indicações para “Fantasy Games” ou até mesmo ajudar comissões técnicas das equipes.

Obrigado !