Aprendizagem de Máquina

Aluno: Allyson Manoel
Login: amnv

Lista de Exercícios 3

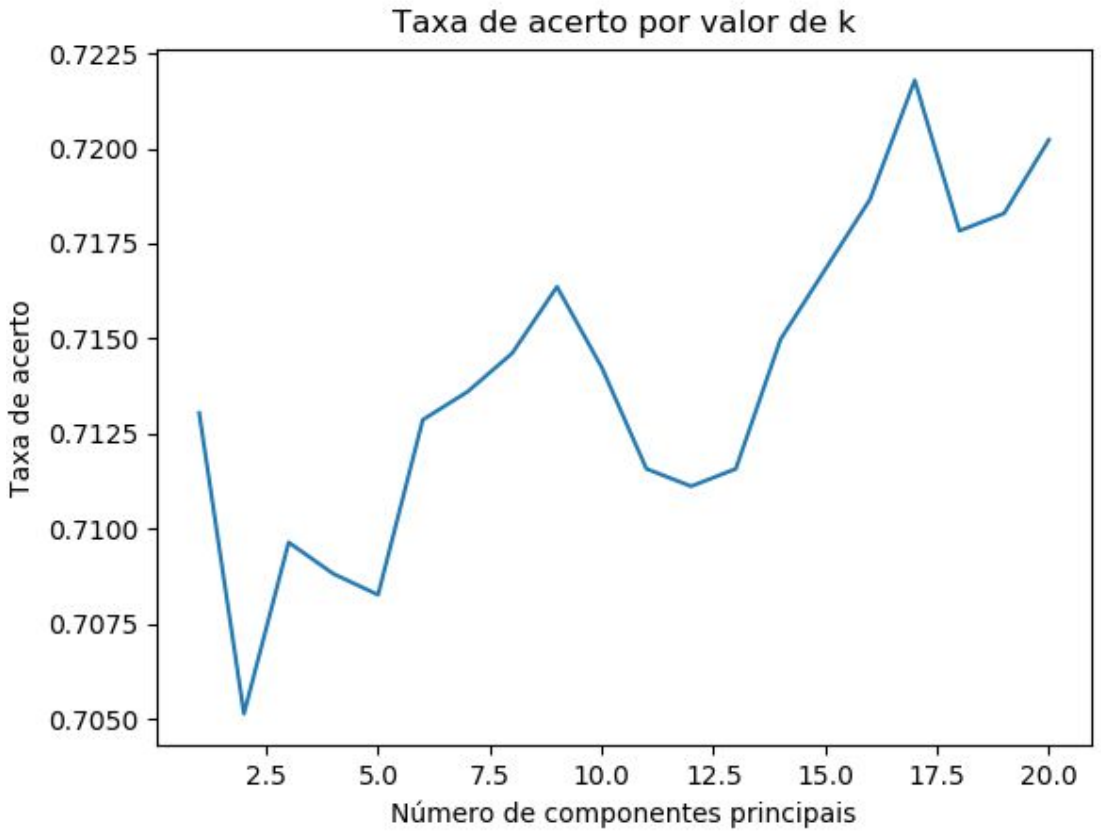
A implementação foi realizada utilizando a linguagem python junto com as bibliotecas pandas para leitura e manipulação dos dados, numpy para lidar com os vetores e cálculos matemáticos, as classe *KNeighborsClassifier* e *MinMaxScaler* do sklearn para utilização do knn e pré processamento dos dados respectivamente e o matplotlib foi utilizado para a visualização dos dados. Os códigos foram inicialmente desenvolvidos utilizando o Jupyter Notebook e posteriormente refatorados em arquivos .py.

As bases utilizadas foram a KC1/software defect prediction e a JM1/software defect prediction. Foi realizado um pré processamento em ambas para localizar e remover dados faltantes e após foi realizados os testes de análise de desempenho. A divisão dos dados foi realizada de forma estratificada e foi utilizado o k-fold cross validation, dividindo os dados em 5 partes.

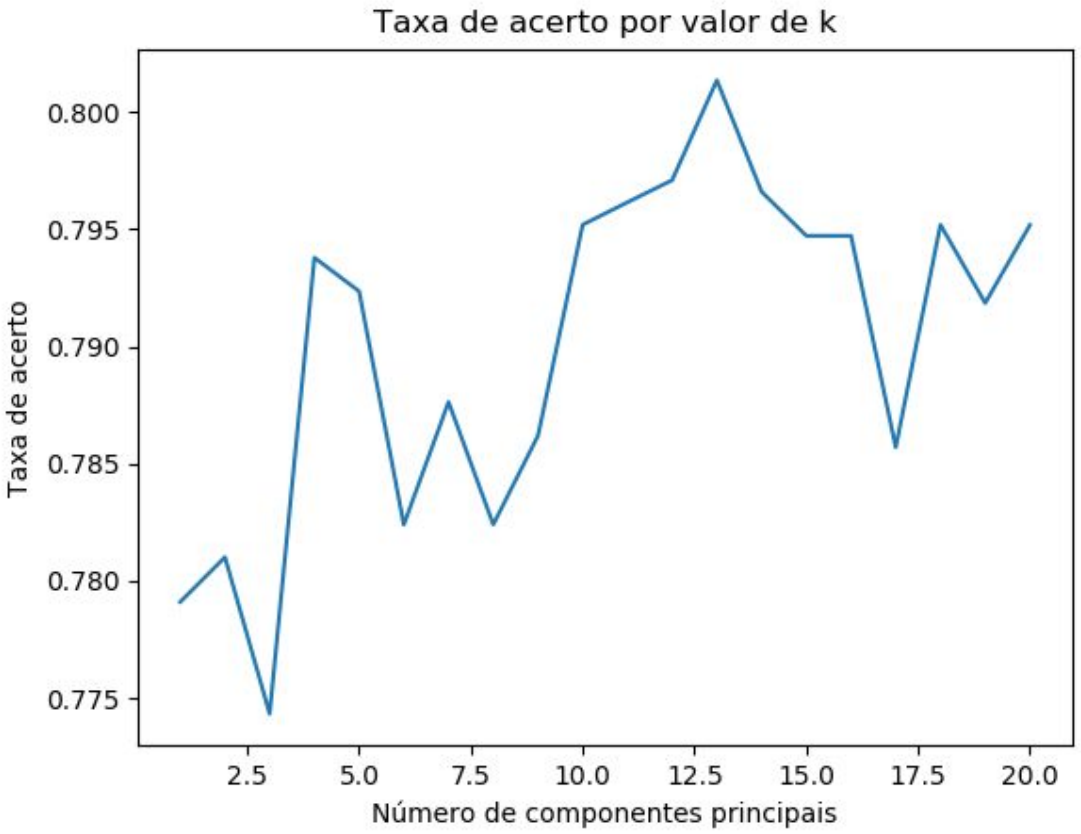
O Knn foi utilizado com k igual a 1 e este valor não foi variado durante os testes. O valor da quantidade de dimensões que os dados tiveram após as transformações variou de 1 até a o número de dimensões menos 1 para o PCA e do número de classes (como as bases eram binárias o número inicial era 2) até a quantidade de dimensões menos 1.

Análise de Componentes Principais (PCA)

Em ambas as bases foi possível observar que a redução na dimensionalidade impactou no desempenho do modelo. Embora não seja uma relação linear, é possível observar que quanto mais próximo da dimensionalidade inicial, ou seja, quantos menos atributos são removidos da base, melhor o resultado apresentado pelo modelo. A variação no desempenho nas duas bases testadas não chegou a 3%.



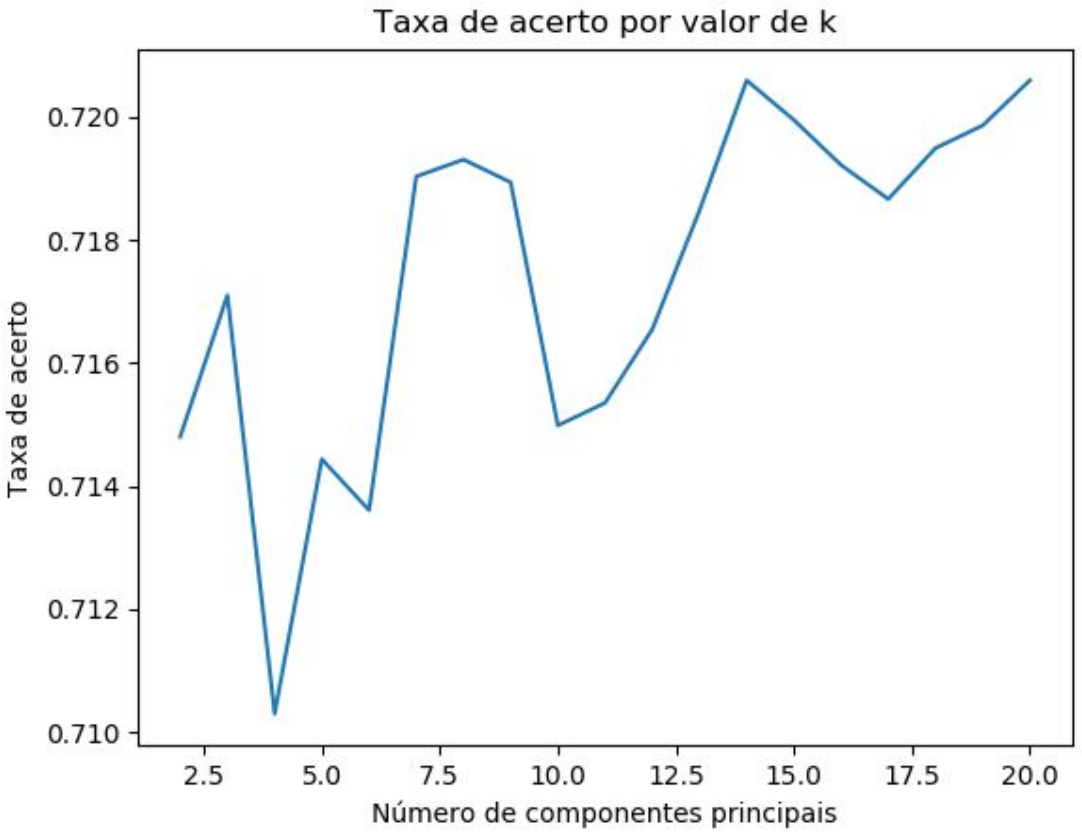
Resultado do KNN após redução de dimensionalidade utilizando PCA na base JM1. Valor de k variando de 1 a 20.



Resultado do KNN após redução de dimensionalidade utilizando PCA na base KC1. Valor de k variando de 1 a 20.

Análise de Discriminante Linear (LDA)

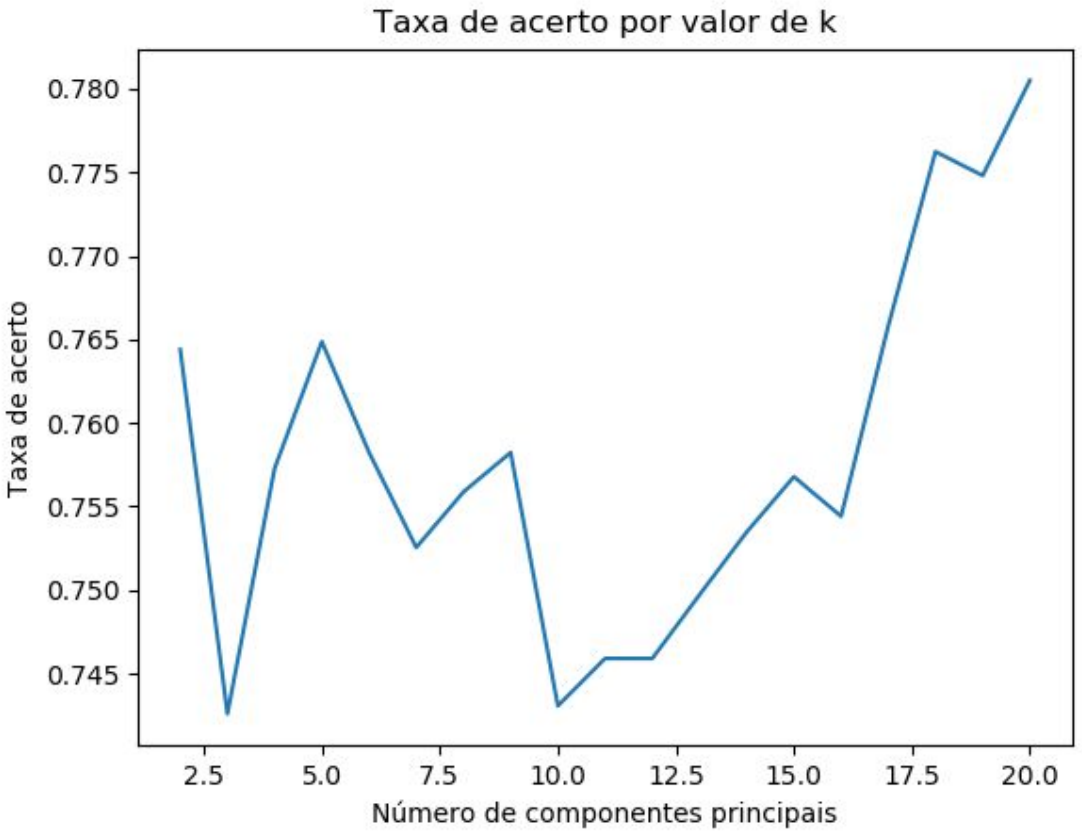
Assim como no PCA, no LDA é possível observar em ambas as bases que quanto mais próximo da dimensionalidade original melhor o desempenho do KNN. A relação entre a redução de dimensões o de desempenho do modelo não é linear, porém é possível observar uma melhora no desempenho ao aumentar a quantidade de dimensões originais. Para a base KC1 o desempenho do modelo variou em no máximo 5% e teve bons resultados para a quantidade mínima de dimensões (2 para ambas as bases) se comparado com o resultado do mesmo modelo quantidade maior de dimensões.



Resultado do experimento utilizando o algoritmo LDA para redução de dimensionalidade na base JM1. Teste realizado utilizando knn com valor de k igual a 1.

Links para as bases

http://promise.site.uottawa.ca/SERepository/datasets/jm1.arff
http://promise.site.uottawa.ca/SERepository/datasets/kc1.arff



Resultado do experimento utilizando o algoritmo LDA para redução de dimensionalidade na base KC1. Teste realizado utilizando knn com valor de k igual a 1.