Predição do Ano de Músicas a Partir de Timbres

André Almeida
RA: 164047
fda.andre@gmail.com

Igor Torrente
RA: 169820
igortorrente@hotmail.com

I. Introdução

Nesse trabalho, estudamos técnicas de aprendizado de máquina para obter um modelo linear que prediz o ano de uma música com base em diversos métricas de timbre, armazenadas em um banco de dados populado com diversas amostras de músicas comerciais entre 1922-2010. Foram estudados diversos algoritmos lineares e subconjuntos de treino para comparação de resultados.
Utilizamos a biblioteca *sklearn* [1] do *Python*, que já contem implementações de diversos algoritmos de aprendizado de máquina, além de funções de métricas e preprocessamento. Foi utilizado para importar os dados do arquivo `csv` a biblioteca *Pandas*. O *Jupyter* foi escolhido como ambiente de desenvolvimento.
Os recurso computacional disponível foi um computador com um processador Intel Core i7-3537U (2.00GHz × 4) e com 7,7 GiB disponíveis de memória RAM.

II. Dados

A base de dados contém no total 499.973 amostras, com o ano da música e 90 atributos, sendo destes: 12 médias de timbre e 78 covariâncias de timbre. O banco de dados foi obtido a partir do LabROSA [2], da Columbia University. Os dados foram divididos como 93,2% para treino e 7,8% para teste. Adicionamos, para facilitar a manipulação dos dados, cabeçalhos às colunas do banco. A primeira coluna de ano foi nomeada `year`, as 12 colunas de médias de timbres `ta01`...`ta12` e as 78 colunas de covariância `tc01`...`tc78`.
Separamos os dados de treino como 80% de treino e 20% para validação. Embaralhamos os dados para evitar qualquer ordenação no banco de dados original, que pudesse deixar algum viés. Os dados do treino foram usados apenas uma vez, após os testes com a validação. Os *scores* e percentagem dos acertos descritos no artigo são referentes ao conjunto de testes.
O ano das músicas é o *y* (ou *f(x)*) e os 90 parâmetros são o *x*. Conforme orientado em [3], no capítulo de *Gradient Descent*, aplicamos uma normalização (`StandardScaler`) que normaliza os dados para não haver maior influência dos que variam mais, assim melhorando a convergência da função. Existe um pico de músicas na década de 2000, levando essa década a ter uma variabilidade maior nos dados. Além disso, acredita-se que devido o avanço das técnicas de gravação e produção sonora durante o tempo, isso permitiu músicas com variedades de timbres mais complexas.

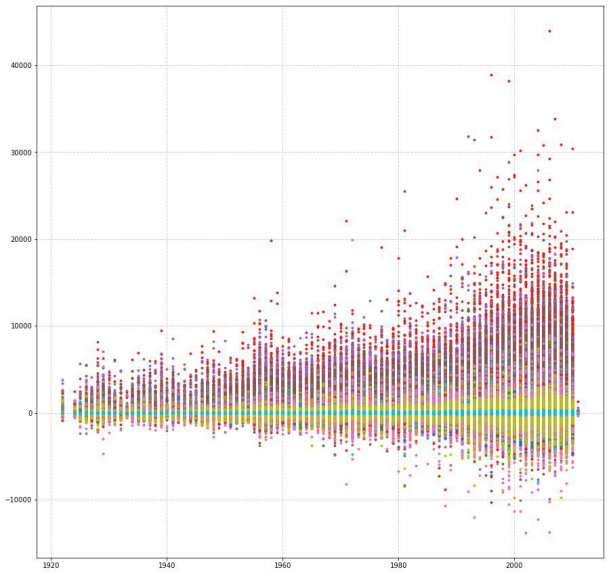

Figura 1: Valores da 90 features (cada uma representada por uma cor) x por ano

III. Soluções propostas

Nessa sessão e nas próximas, *score* é a métrica de acerto do dado definida como o coeficiente de determinação R^2. O coeficiente R^2 é calculado como $(1 - u/v)$, onde u é a soma residual dos quadrados $(y_{real} - y_{predito})^2$ e v é a soma total dos quadrados $(y_{real} - media(y_{real})^2$, para todos os y. O melhor valor possível é 1, e pode ser arbitrariamente negativo. [4]

A. Regressão linear com descida estocástica de gradiente

Utilizando o módulo `linear_model.SGDRegressor`, aplicamos uma decida de gradiente na função de custo de uma regressão linear e obtivemos uma função. Utilizamos uma taxa de aprendizado de 10^{-4} e o número máximo de iterações igual a 100, e obtivemos o score mais alto no conjunto de validação: 0.2331. Os outros parâmetros não foram alterados pois não contribuíam com alterações nos resultados. Foram testados outros valores, mas que não convergiram tão bem:

`eta0`	`max_iter`	score
10^{-6}	50	-15775.02
10^{-3}	200	0.2293
10^{-5}	150	0.2297

No conjunto de testes, o score final foi de **0.2293** com **5.18%** de acerto.

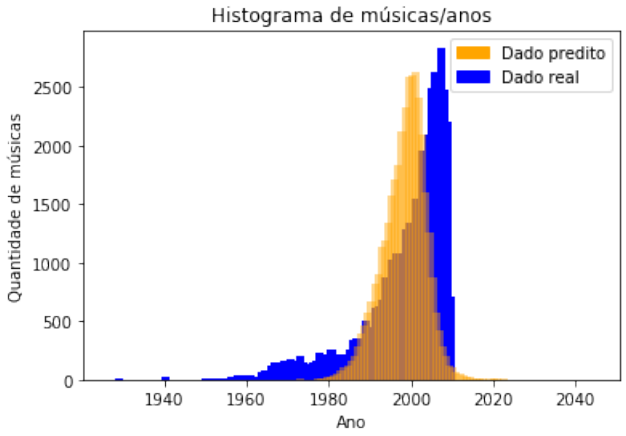

Figura 2: histograma da primeira solução

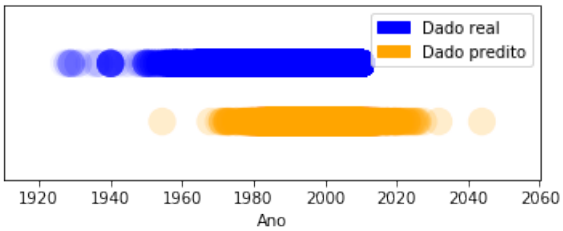

Figura 3: distribuição dos anos nos conjuntos

B. Regressão linear com Lasso

Lasso [5] é um método linear que usa o erro L1 (erro absoluto) como regularizador. Utilizamos o `linear_model.Lasso` em nosso código, mudando apenas o parâmetro `alpha`, deixando com o valor `0.0001`. Lasso é considerado como um modelo mais poderoso, por regularizar os dados previamente antes de aplicar a regressão linear. Obtivemos como score no teste **0.2292** e acerto **5.17%**.

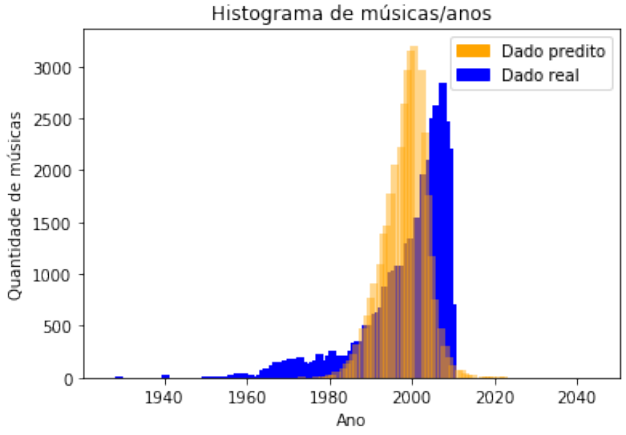

Figura 4: histograma da segunda solução

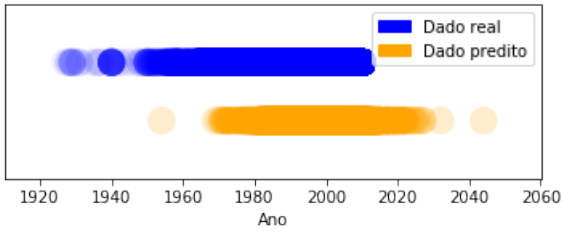

Figura 5: distribuição dos anos nos conjuntos

C. Regressão linear com Equação Normal

Para testarmos a regressão linear com a decida de equação normal, usamos a função `linear_model.LinearRegression`, que já implementa por padrão a equação normal para minimizar o erro. Mantemos o parâmetro `Normalize` desativado, já que não ele promoveu nenhuma alteração no resultado. Usamos os conjuntos de dados sem a regularização, obtendo um score ligeiramente maior (0.001%). Os resultados finais foram: score **0.2297** e acerto de **5.24%**.

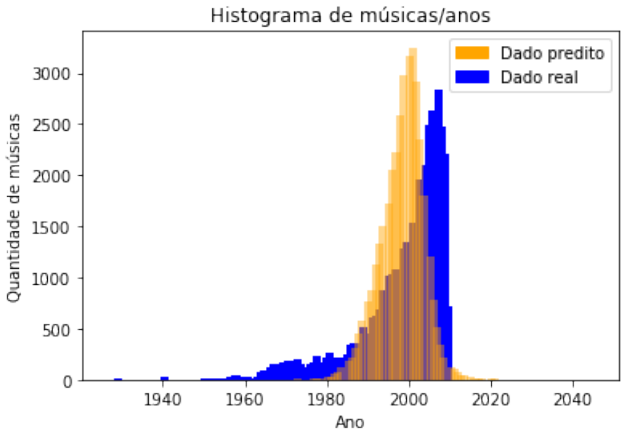

Figura 6: histograma da segunda solução

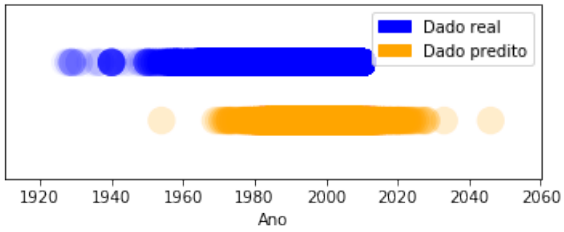

Figura 7: distribuição dos anos nos conjuntos

Na figura 8 podemos ver a diferença entre o valor real e o valor predito no nosso modelo. Um modelo com *overfitting* apresentaria uma reta perfeita crescente.

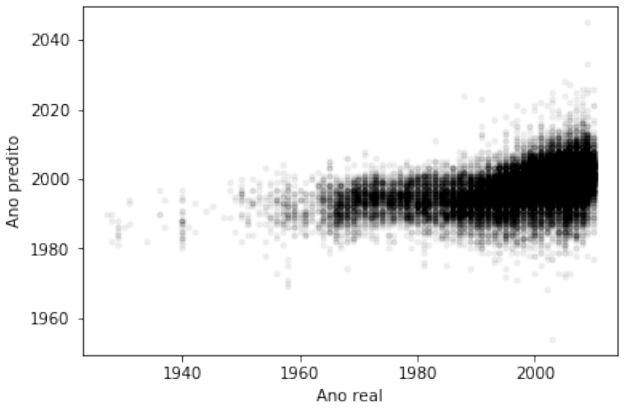

Figura 8: anos preditos x anos reais

IV. Descida de gradiente

Tanto os modelos automáticos (`linear_model.Lasso` e `linear_model.LinearRegression`) tanto o modelo manual (`linear_model.SGDRegressor`) tiveram resultados muito próximos da descida de gradiente/equação normal, já que observamos scores muito próximos.

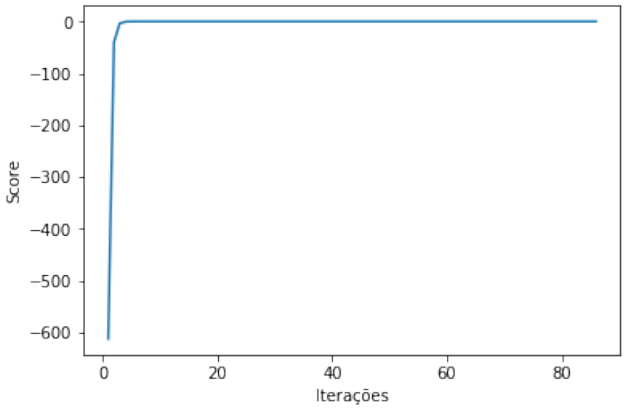

Figura 9: Score x Número de iterações (`SGDRegressor`)

V. Resultados

Nos gráficos [2-4-6] temos um histograma comparativo da predição do nosso modelos (laranja) e os dados reais (azul), que mostra que nossos modelos não predizem a quantidade correta de músicas nos seu respectivos anos. Nosso modelo se assemelha a uma gaussiana um pouco deslocada para a esquerda.
Nos gráficos [3-5-7] temos a concentração das predições dos modelos (laranja) e dos dados reais (azul) baseado na tonalidade da cor, demonstrando uma relativamente alta predição do nosso modelo depois do ano de 2011 e uma baixa predição nos anos entre 1922 a 1965. O modelo parece ”descolar” os anos cerca de três décadas à frente.

A. Descida de gradiente estocástica

Como visto anteriormente na sessão de resultados, conseguimos um score e uma precisão baixa tanto nas validações quanto no teste e a proporção de treino e validação não influenciou a precisão dos modelos, indicando que ele é muito simples e não se adequa corretamente aos nossos dados, observa-se um claro underfiting.

B. Lasso e Normal

A conclusão dada a descida de gradiente estocástica se aplica ao método de Lasso e a Normal com resultados de score e precisão também muito próximos, podendo ser observado pela similaridade os gráficos de todos os modelos obtidos com as três técnicas.

C. Análise dos erros

Analisamos os erros no `LinearRegression`. Das 34384 predições erradas, 13922 (40,4%) erraram para um ano maior do que o real, e 20462 (58,7%) erraram para um ano menor. Dos erros para cima, a média foi de 8.89 anos, e para baixo, de 6.05 anos. Os erros por décadas podem ser vistos na tabela abaixo:

Década	**Erro**
1920	12
1930	704
1940	19949
1950	8055
1960	2876
1970	1694
1980	838
1990	207
2000	39
2010	10

Uma possível explicação seria o fato do grande aglomerado de músicas a partir do século XXI no banco criarem um viés para ”cima” e a falta de dados antigos tornam eles menos precisos.

VI. Conclusão

Nenhum modelo linear foi capaz de criar funções que predizem corretamente o ano a partir da base de dados usadas, mas o adequado entre eles foi a Regressão linear com Equação Normal (`linear_model.LinearRegression`), com um score **0.2297** e acerto de **5.24%**.

VII. Estudos futuros

Precisamos de mais estudo sobre os dados para descobrimos mais o comportamento e correlação entre eles, uso de técnicas de pré-processamento para removermos dados pouco relevantes e inter-relacionados entre si. Também é necessário mais hardware e modelos mais complexos (árvore de decisão, polinomial) que se ajustam melhor ao problema. Uma possível alternativa seria alterar

References

[1] Scikit-learn: Machine Learning in Python, acessado em 31 de Agosto de 2017, em `http://scikit-learn.org/stable/index.html`
[2] Million Song Dataset, em: http://labrosa.ee.columbia.edu/millionsong/
[3] Geron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.*

[4] Scikit-learn: sklearn.linear_model.LinearRegression, acessado em 31 de Agosto de 2017, em http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
[5] Scikit-learn: sklearn.linear_model.Lasso, acessado em 31 de Agosto de 2017, em http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html