**Análise do grau de “danceability” musical usando regressão logística no Spotify**

*Carla Kim Gaieski, Giulia Braga Passarelli, Rebeca Moreno dos Santos*

**Introdução**

Imagine que você é um músico e quer compor uma música, porém quer que ela seja calma, qual a probabilidade de você usar o tom menor e qual nota provavelmente usaria? O objetivo desse projeto é responder esta pergunta prevendo qual seria a decisão que você tomaria nesta situação. Assim, temos que comparar os tons e as notas com o quão dançável as músicas são.

Nesse caso, o método escolhido foi a regressão pois é uma das ferramentas mais utilizadas para investigar a relação entre variáveis na estatística. Ela opera equacionando a variação de uma variável dependente por conta da alteração de uma independente. Para isso, há dois modelos que são mais comuns: a regressão linear e a regressão logística. A primeira é considerada quando a variável de saída é quantitativa, pois o gráfico apresenta um comportamento mais linear sendo, geralmente, mais fácil de encaixar uma função em cima. Já o segundo é próprio para a análise de uma variável de saída categórica, que não se porta da mesma forma que o último. Como queremos analisar as notas e os tons musicais (variáveis qualitativas), elas devem ser a saída em uma regressão logística.

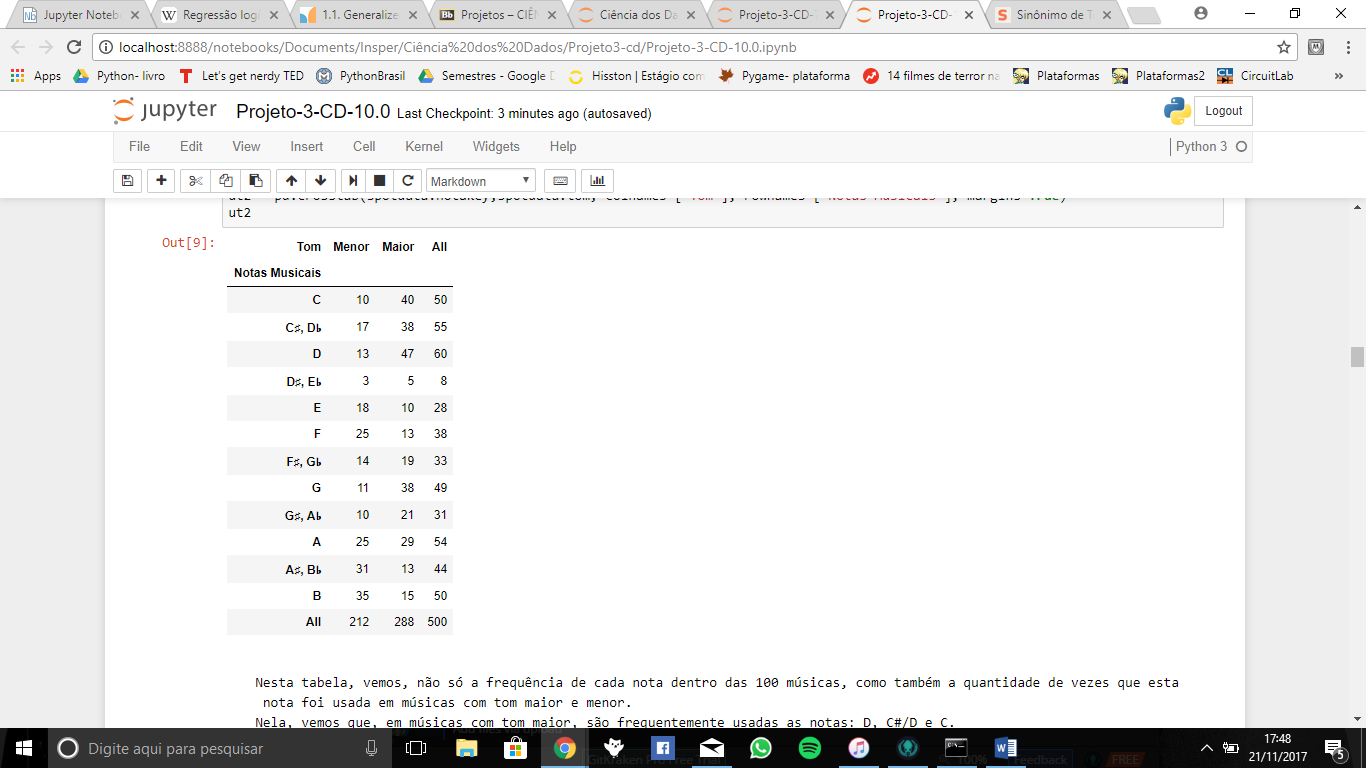
A regressão logística possui uma função para a probabilidade de um ponto que pode ser definida como: .

**Procedimento:**

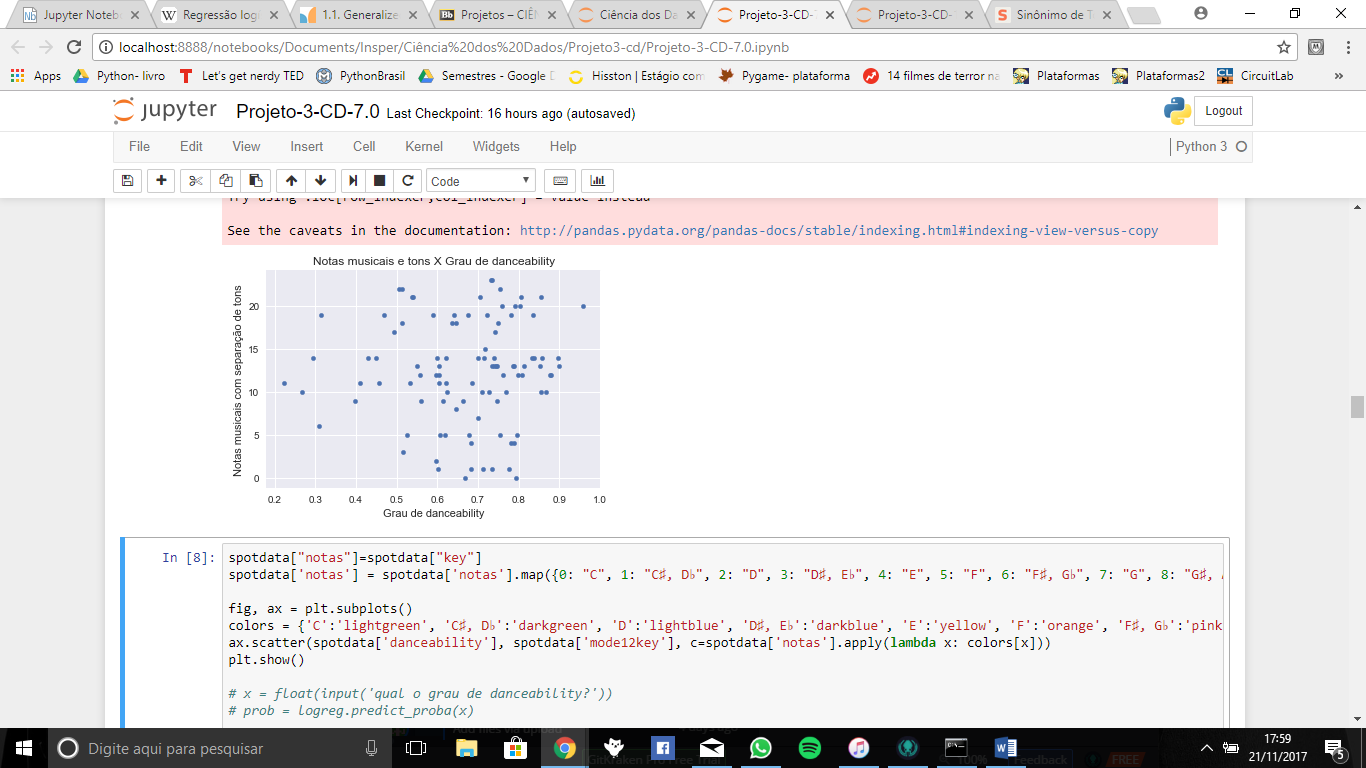
O Spotify é um serviço de música digital com um acervo de milhões de músicas, de forma a permitir o acesso, a qualquer momento, de uma coleção imensa de canções. Por conta disso, o projeto utiliza a base de dados do Spotify para realizar essa análise. Coletamos uma amostra contendo 500 músicas por meio do dataset “Spotify Song Attributes” do site kaggle.com, utilizando as colunas: “danceability” (valores contínuos de 0 a 1), “key” (notas- valores discretos de 0 a 11, sendo 0 a nota C e 11 a nota B -temos do inglês) e “mode” (tom- valor binário-, 0 -tom menor- ou 1 -tom maior).

Podemos relacionar tom e notas musicais para realizar a análise pois cada nota possui uma tonalidade correspondente que retorna um certo grau de “danceability”. E, assim, como estamos considerando a “danceability” das músicas, as duas variáveis possuem uma dependência nesse aspecto. Para realizar essa união de variáveis, criamos uma nova coluna no dataframe (chamada de “mode12key”) que calcula: . O motivo para multiplicarmos o tom por 12 é para ele possuir um intervalo que se assemelhe ao das notas.

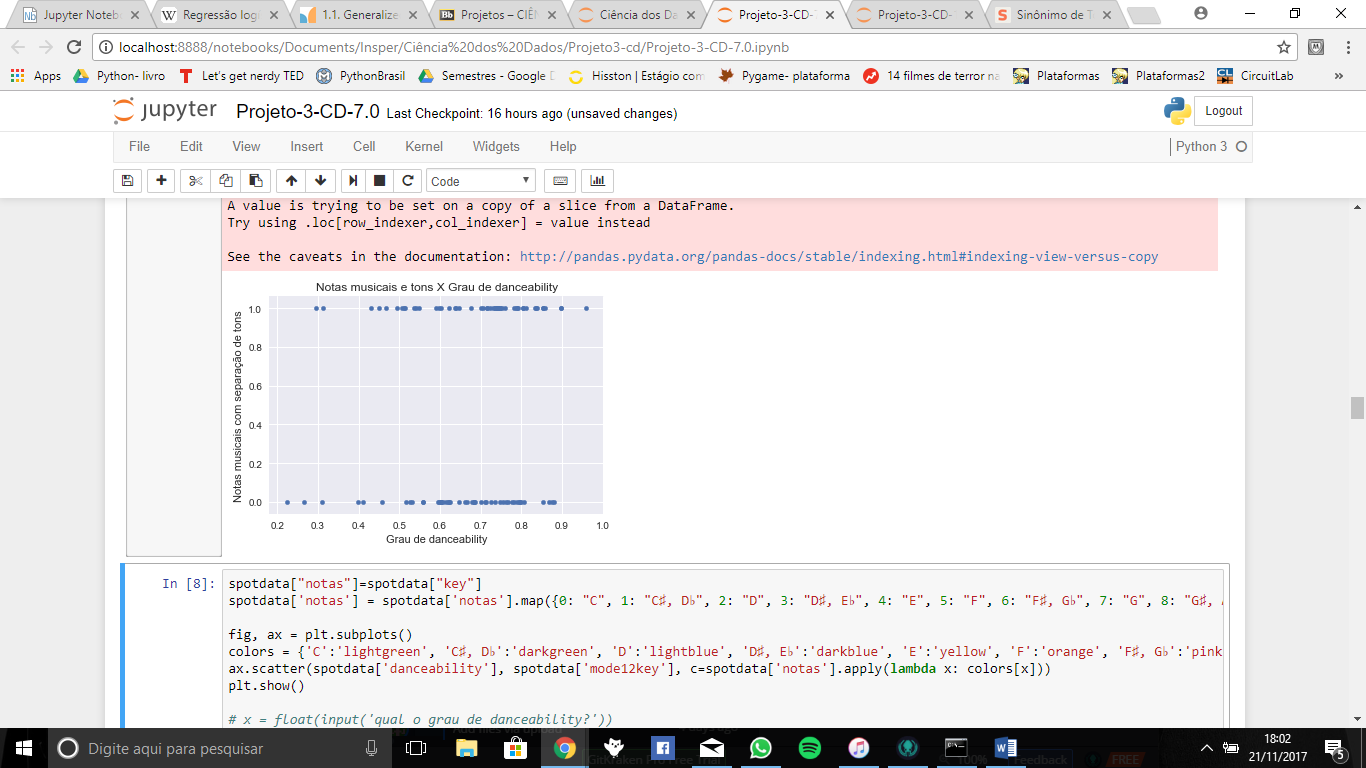
Abaixo está uma tabela com a frequência de cada nota para diferentes tonalidades (tom maior ou tom menor):



Quando geramos um gráfico que relaciona a nova coluna criada no dataframe com a coluna de “danceability”, obtemos:



Para realizar uma regressão logística, o ideal (pois possui maior probabilidade de acerto) seria possuir uma variável dependente binária. Assim, como o “mode12key” é categórico, podemos agrupar os valores que estão acima ou abaixo da média em 0 ou 1, respectivamente, para facilitar a regressão. Assim, geramos o gráfico a seguir de “mode12key” por “danceability”:



Por conta do agrupamento que realizamos, muitos dos dados são perdidos, especialmente das notas, já que o tom da música é um valor binário. Assim, a regressão é obtida com segregação de notas. Para fazer a regressão usufruímos da função LogisticRegression.predict() dentro de sklearn.linear\_model. Ela calcula uma previsão da localização de cada ponto a partir do gráfico anterior. Com isso, possuímos os seguintes resultados:

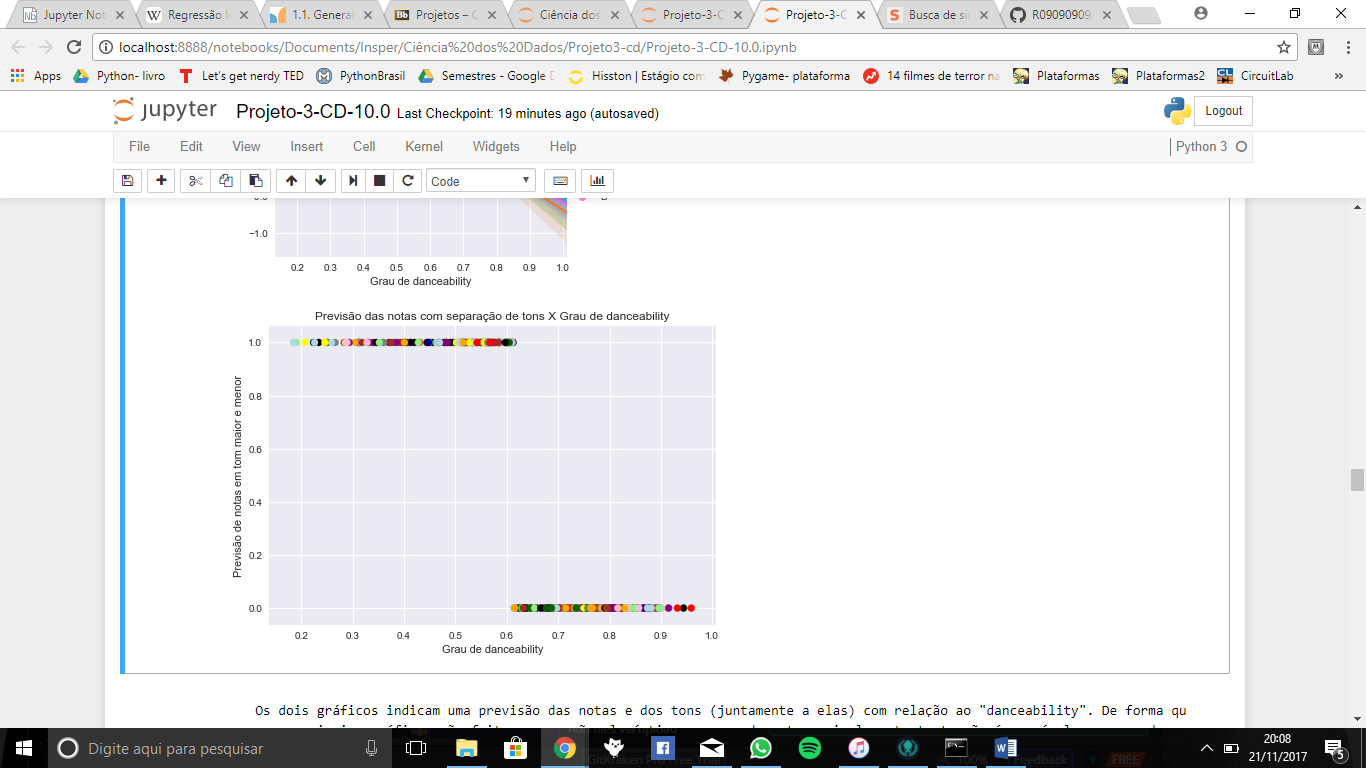
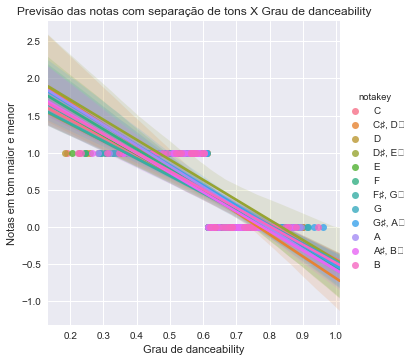
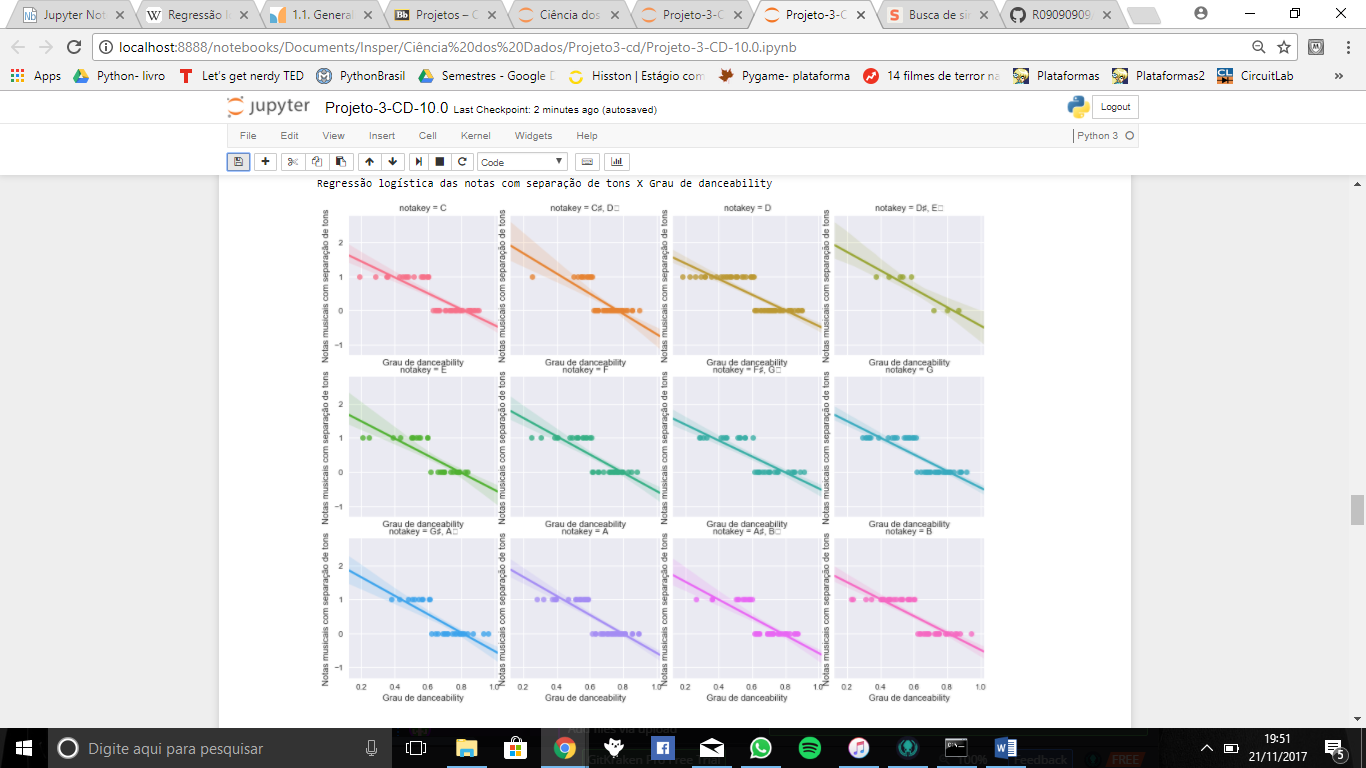
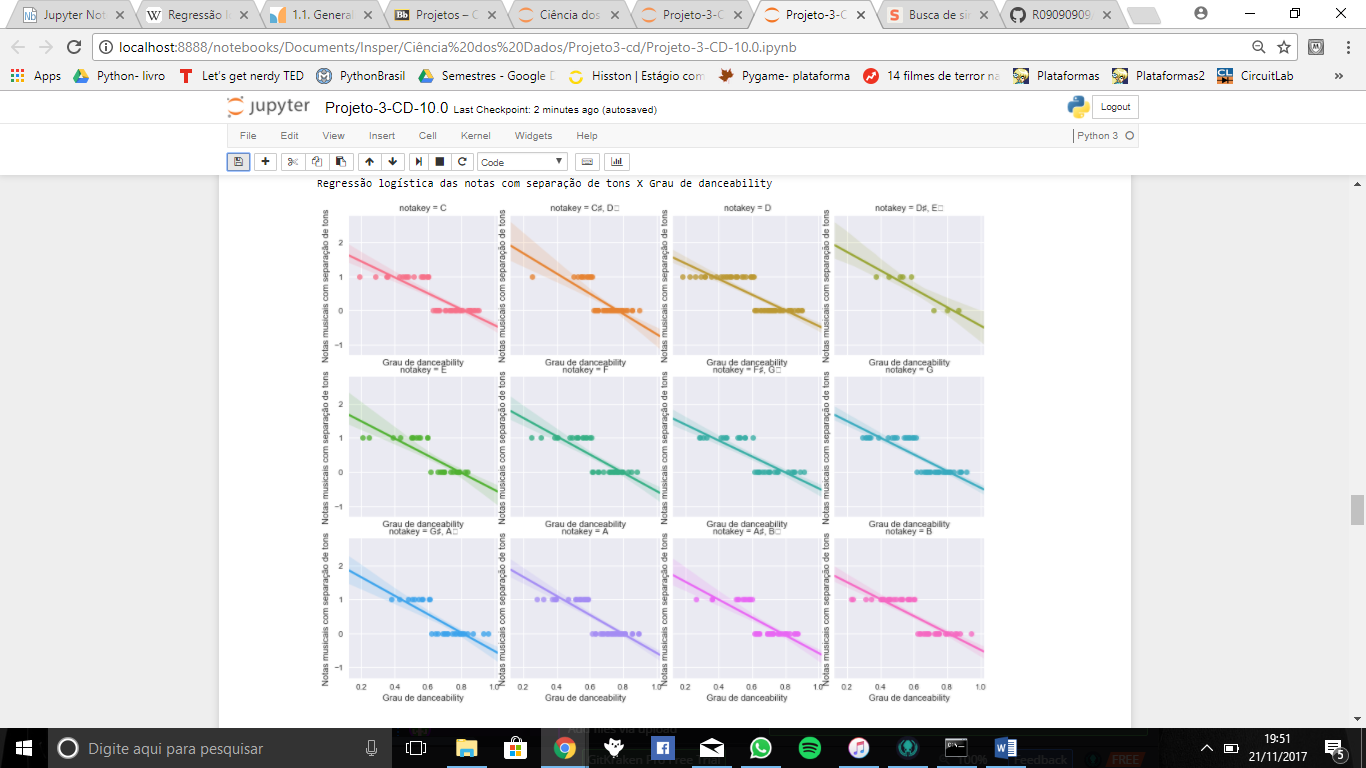
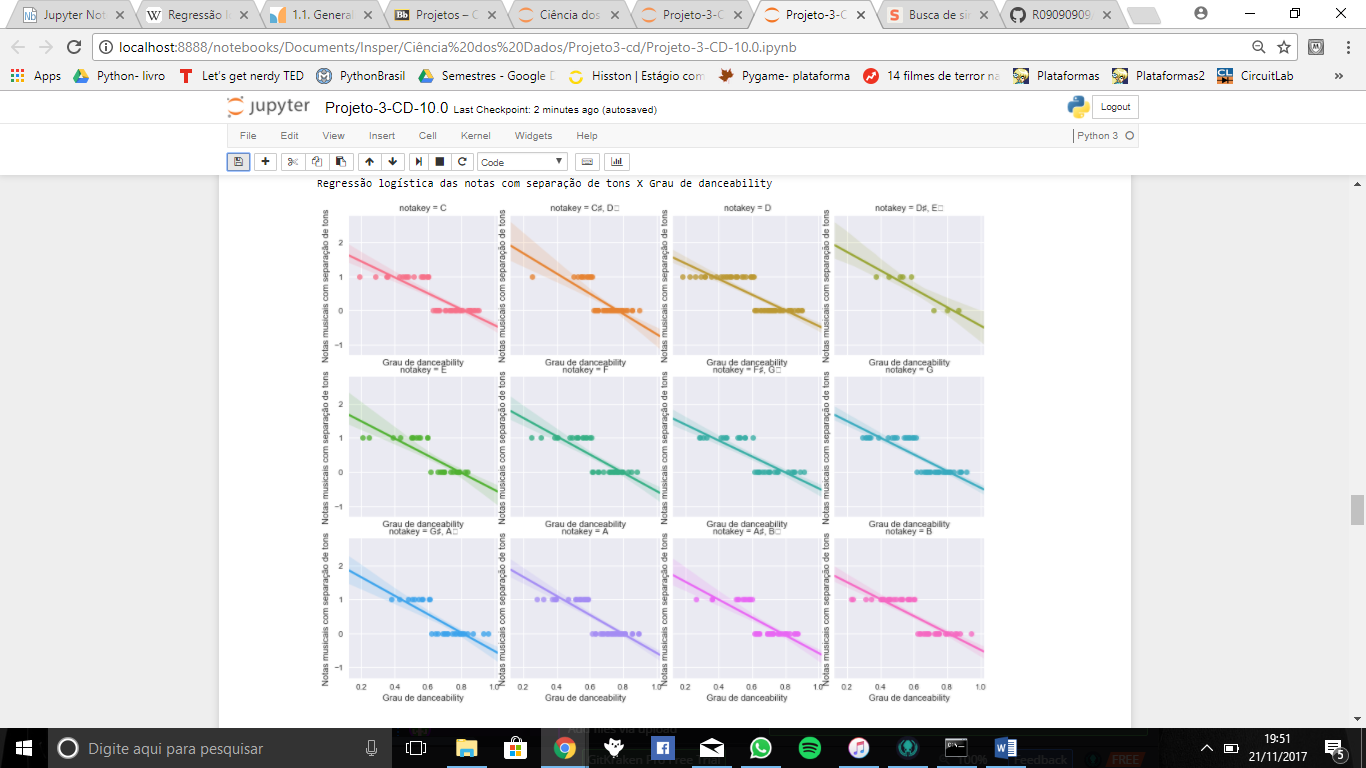
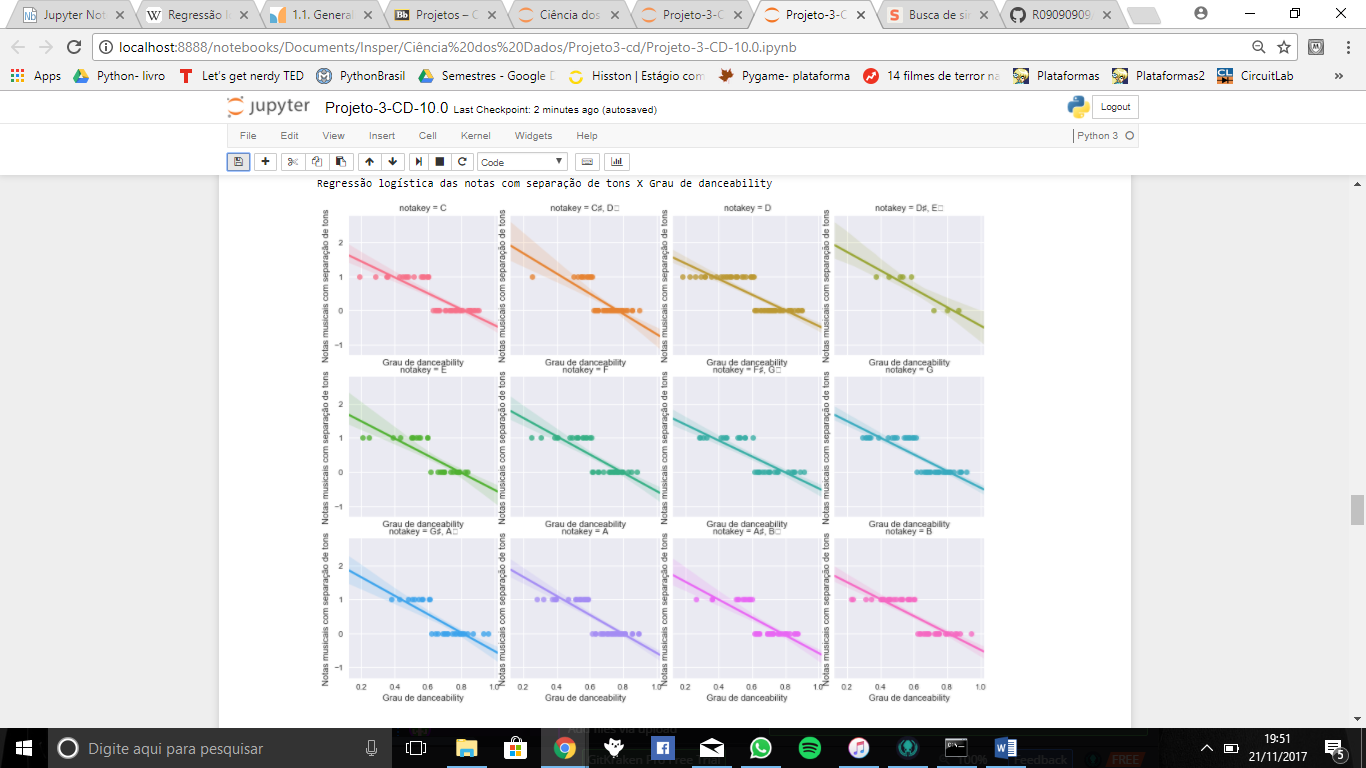
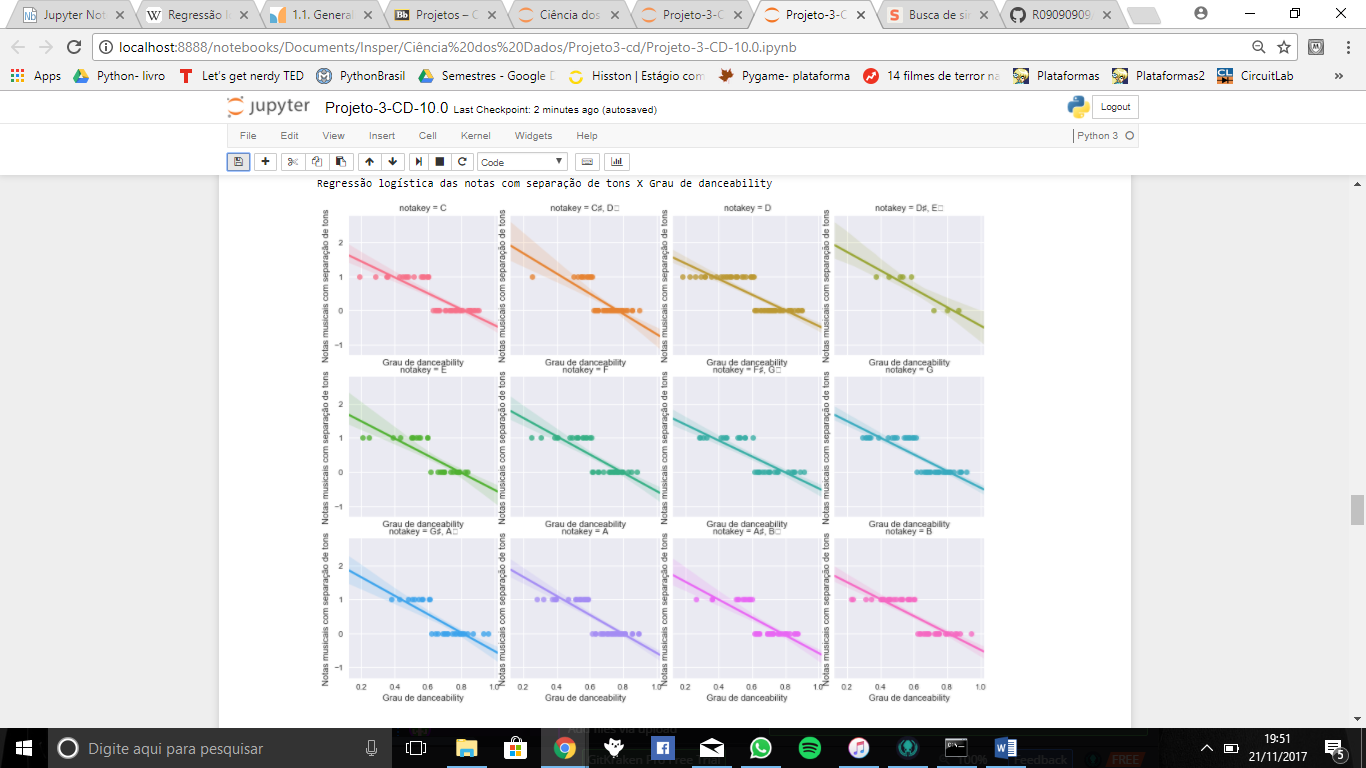
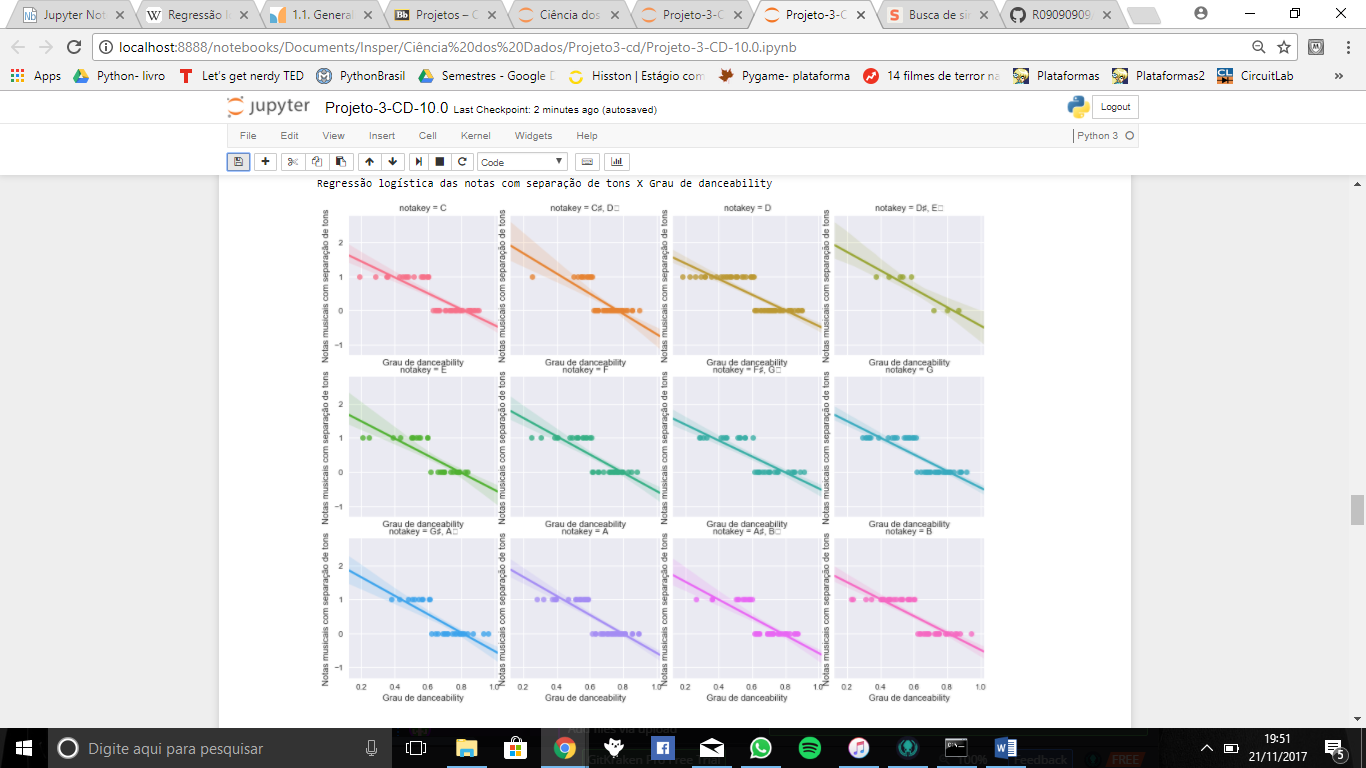
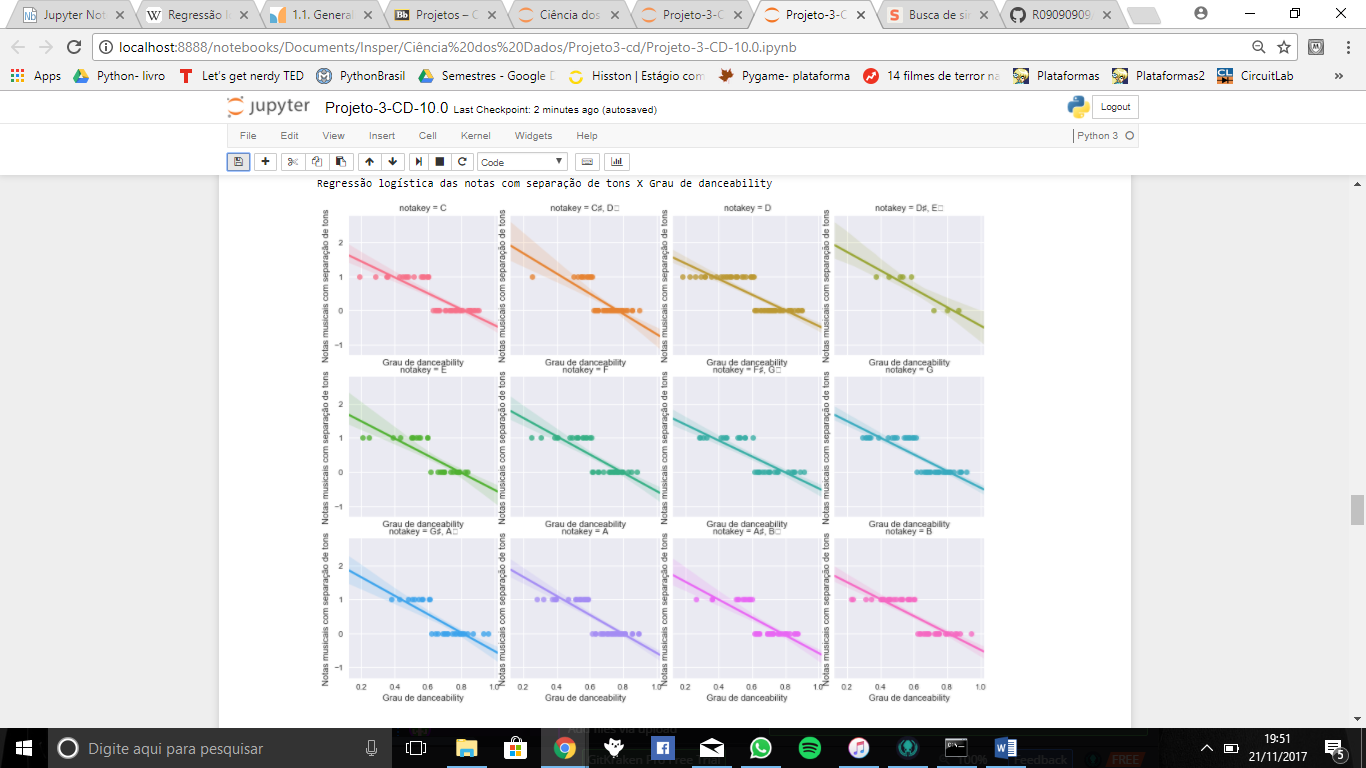
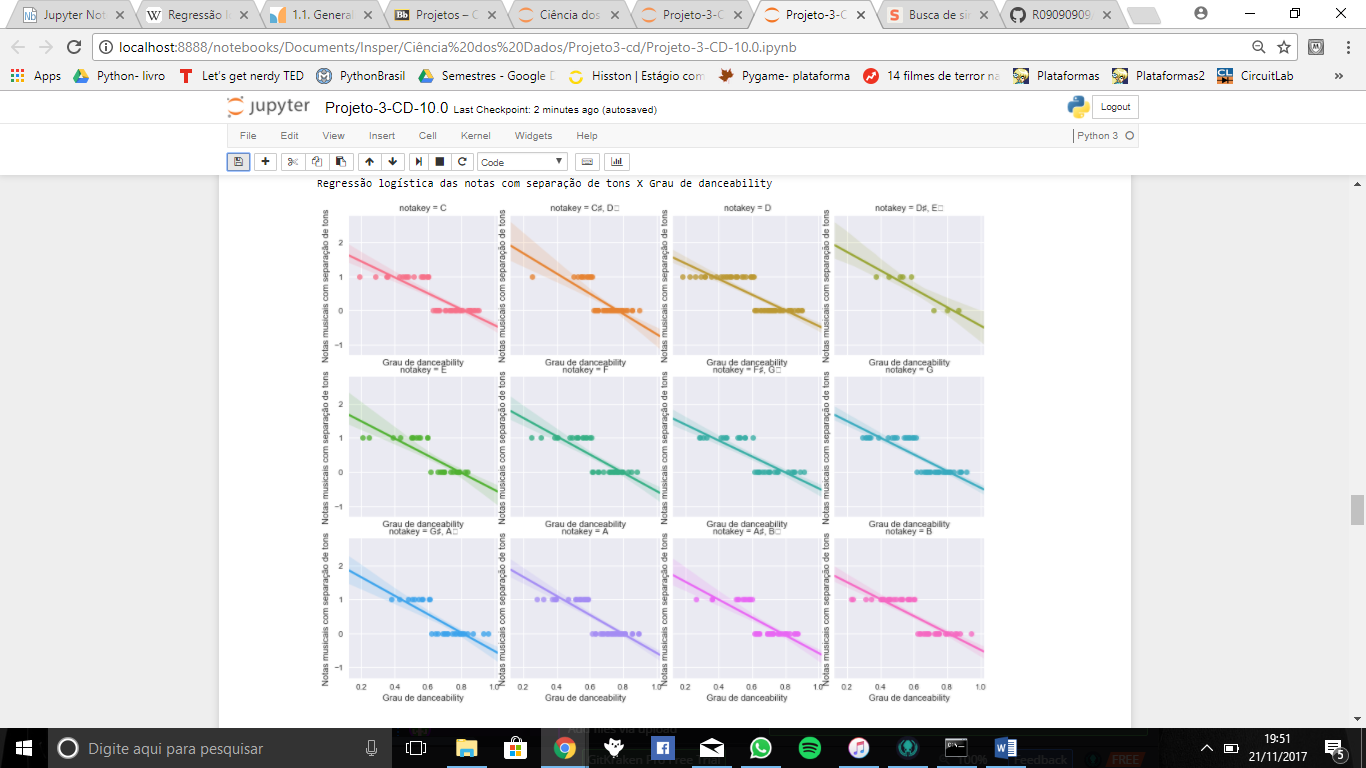
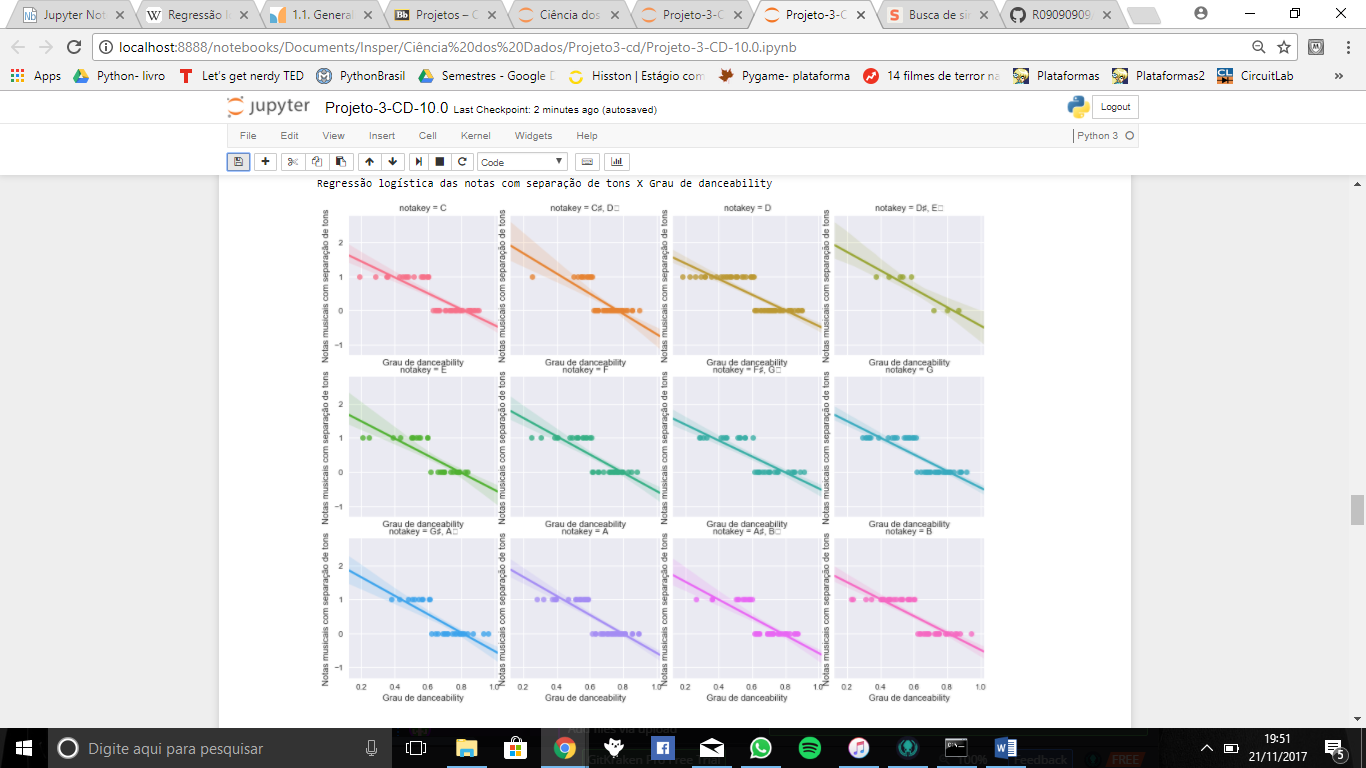


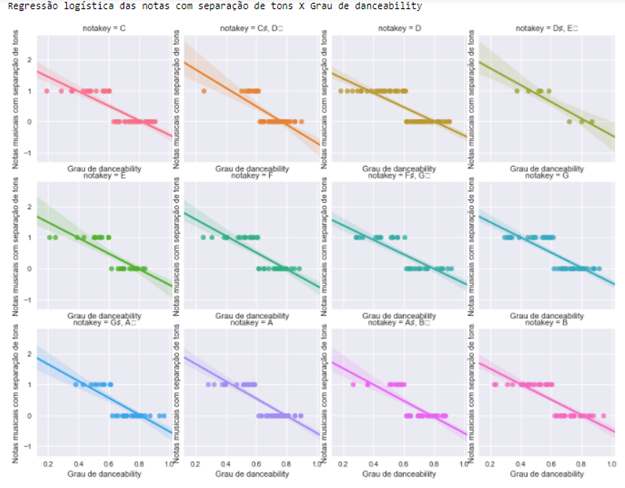
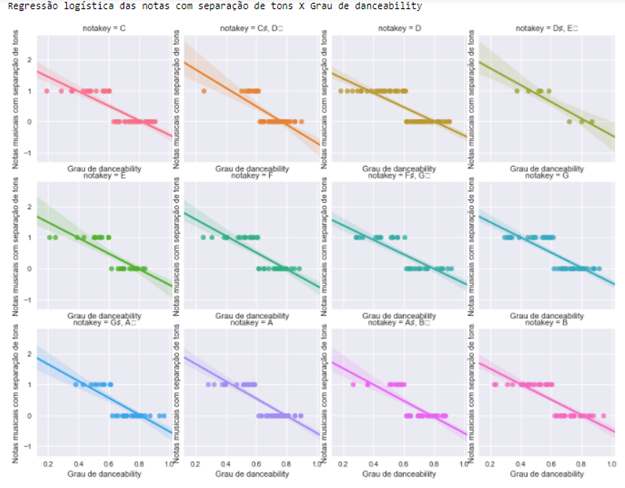
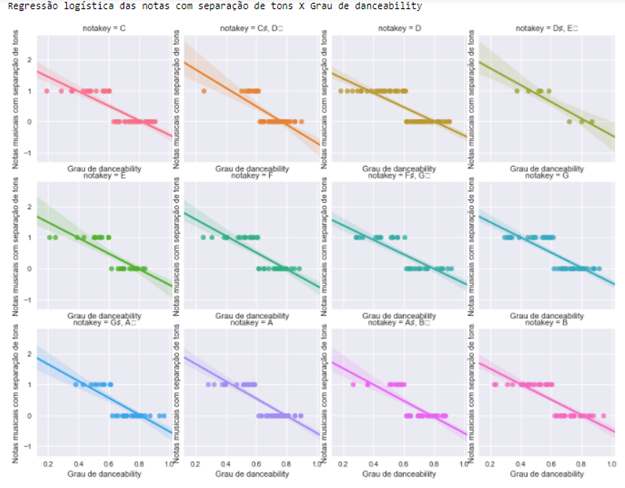
Gráfico de previsão com a função da regressão:



Os pontos de cada cor representam uma nota diferente. Como a análise para cada nota é praticamente impossível, geramos um gráfico por nota a seguir:

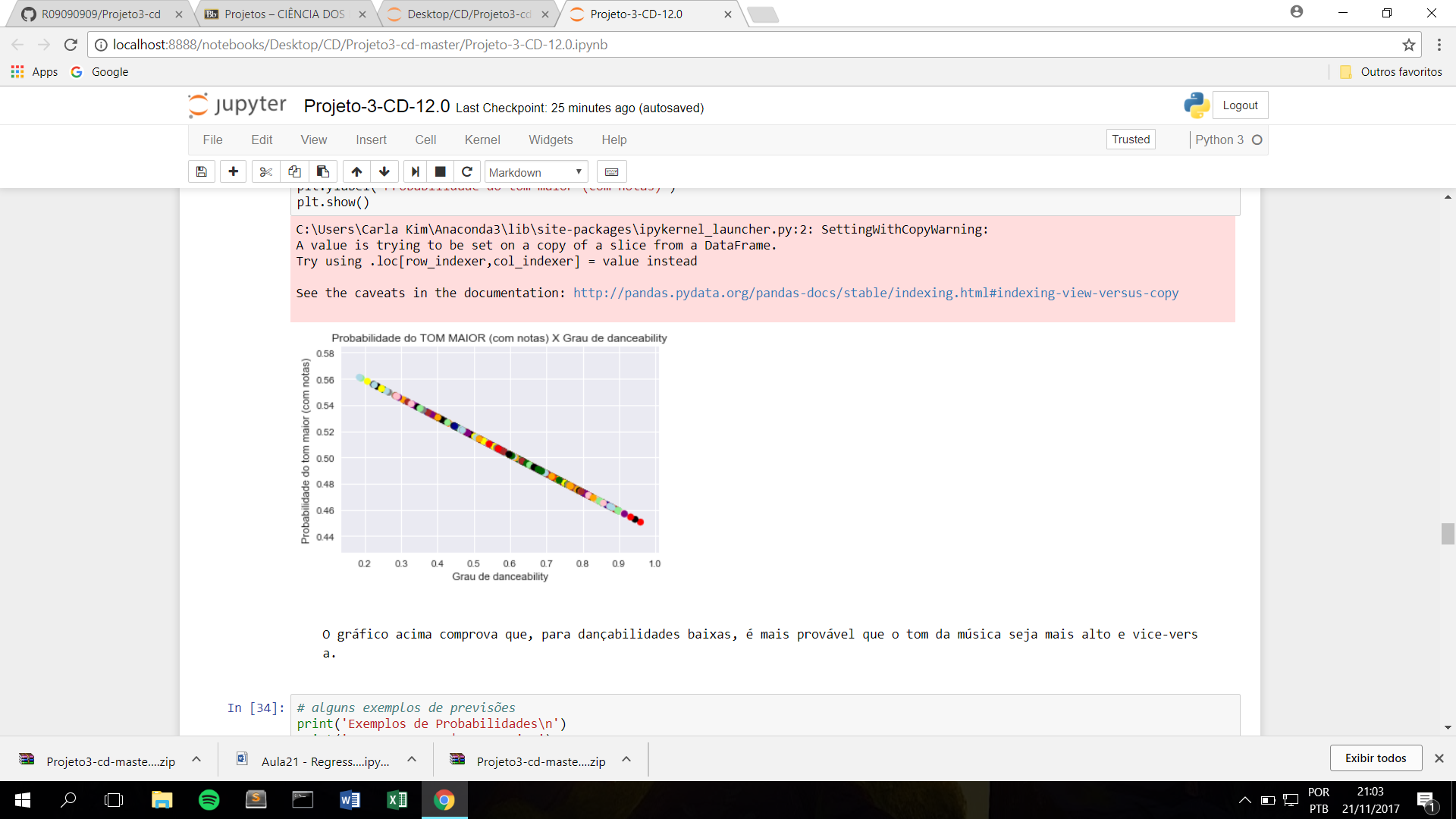
   

A partir dos gráficos acima concluímos que, independentemente da nota considerada, a previsão declara uma probabilidade maior de, para um “danceability” baixo, o tom e as notas seriam maiores do que para um “danceability” mais alto. A "mancha" mais clara que há em algumas regressões nos diz que, como temos poucos pontos do gráfico, há uma margem de erro. Conseguimos ver, também, a quantidade de cada nota em tom maior e menor por cada grau de danceability.

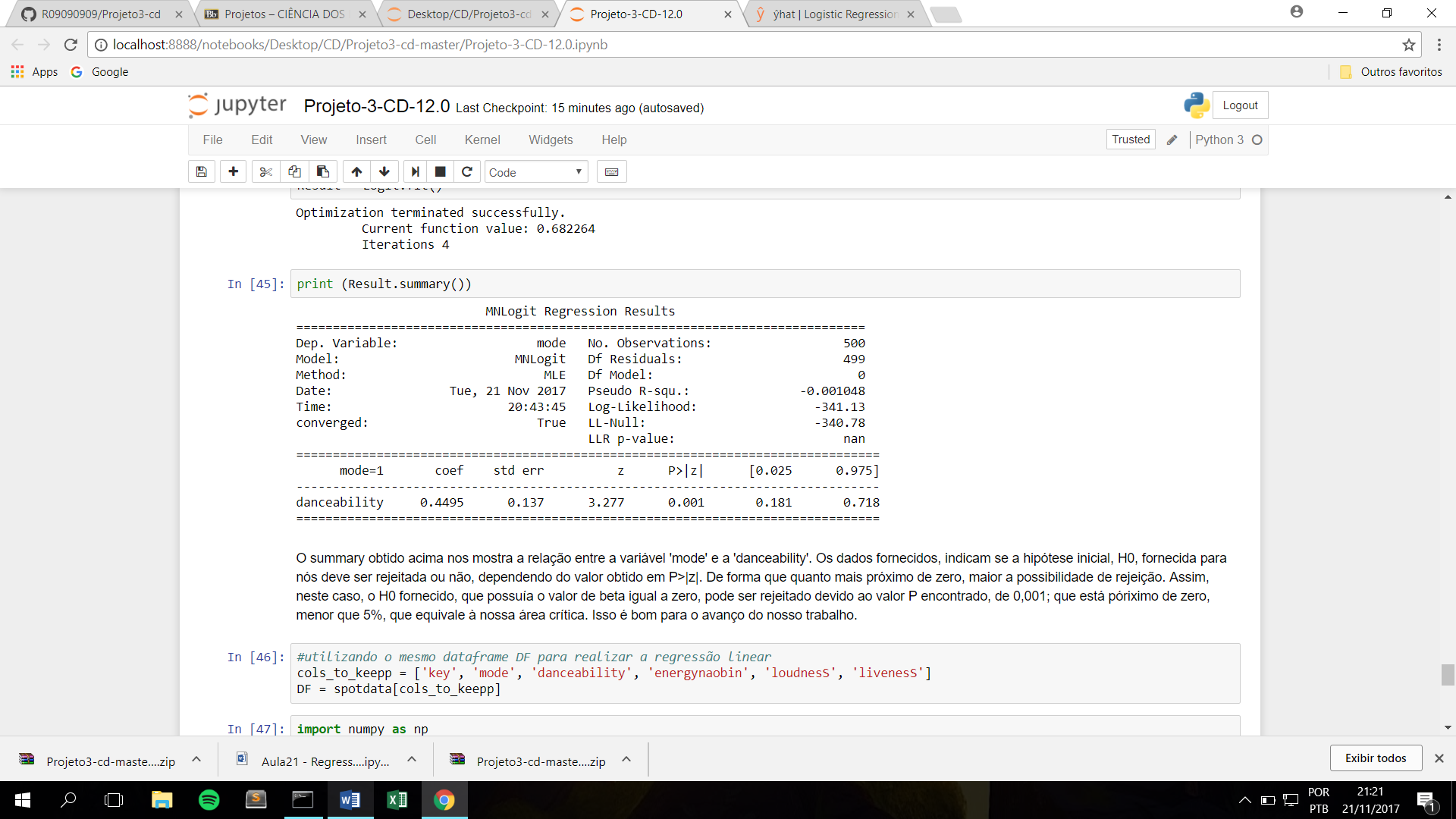
Gerando um gráfico apenas da regressão para as diferentes notas, com a previsão apenas para o tom, temos:



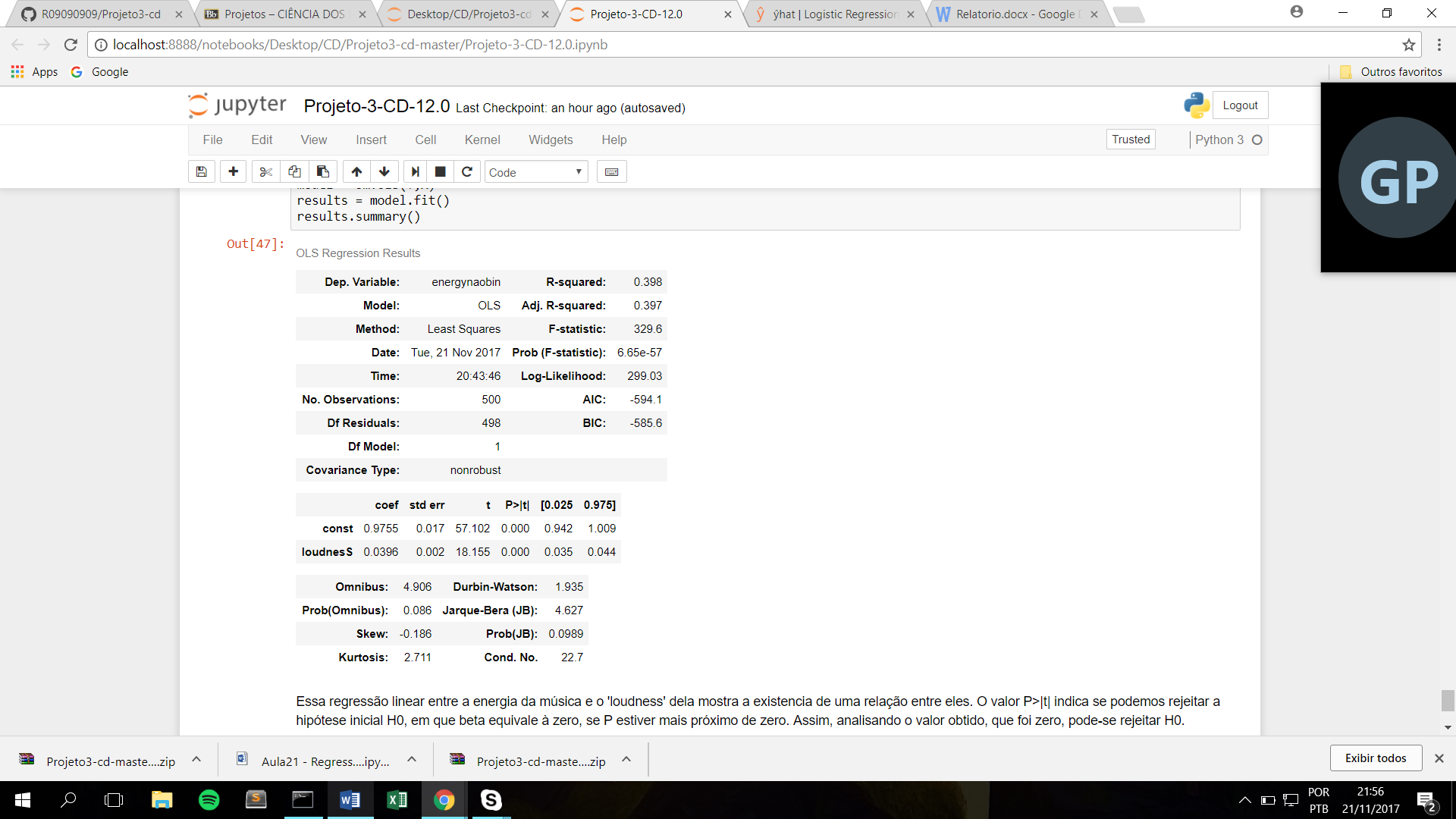
Assim, comprovamos a tendência observada para o “mode12key” porém para o tom da música.

Criamos um novo dataframe, contendo todas as variáveis que utilizaríamos para realizar a regressão logística e posteriormente, na linear. As variáveis eram: 'mode' , 'danceability', 'energynaobin', 'loudnesS', 'livenesS', ou seja, tom, o quão dançável uma música é, energia de uma música descrita de maneira não binária, o quão alto e alegre uma música pode ser, respectivamente. Assim, poderíamos descobrir a tonalidade de uma música baseado nos outros dados. O resultado obtido no valor P nos informaria se a primeira hipótese, ou seja, de que o valor de beta era igual a zero, poderia ser rejeitada ou não.

No primeiro caso, de regressão logística, a variável utilizada para descobrir o tom (‘mode’), foi a ‘danceability’. Ambos aspectos estão bem relacionados de forma que obtivemos um valor P, em , próximo de zero, 0,001. Isso torna-se bom, pois significa que a primeira hipótese pode ser rejeitada e assim, podemos melhorar o sistema, com base em nossos dados. A imagem abaixo expõe todos os resultados da regressão.



A regressão linear realizada nos permitiu explicar a energia de uma música, baseado na variável ‘loudness’. Analisando o valor P, da mesma maneira que a regressão anterior, se seu valor for menor que 5%, H0, a hipótese inicial, poderia ser descartada. A tabela abaixo mostra que o valor P encontrado é 0. Com isso, pode-se dizer que há uma conexão entre as duas variáveis, de forma que podemos melhorar nosso sistema.



Infelizmente não foi possível relacionar todas as variáveis adicionadas ao nosso dataframe DF, das regressões, pois o data frame utilizado do Spotify, está com nossos dados contidos de maneira que não favoreciam os resultados obtidos, como também não justificavam, nem correspondiam com aquilo apresentado anteriormente. Ou seja, quando mais de uma variável era adicionada à regressão, os valores de P aumentavam de forma absurda a ponto de não fazerem sentido ao que havia sido apresentado. Porém, quando cada uma era aplicada separadamente, algumas ainda conseguiam valores de P mais compreensíveis e verossímeis, mas outros ainda permaneciam altos. Mas não poderiam ser utilizadas sem uma correlação, pois não teria sentido à informação que queríamos transmitir.

Com isso, escolhemos as variáveis que mais se relacionam ao tom de uma música, que nos ajudariam a chegar em nossa resposta. No entanto, mesmo não sendo possível chegar exatamente à conclusão esperada, podemos, com os dados obtidos e o código, chegar em uma aproximação da nota musical e com precisão no tom, somente conhecendo a ‘danceability’, que for fornecida.

Para uma futura implementação e continuação do projeto, torna-se interessante nos aprofundarmos mais no comportamento do dataset utilizado, para realmente checar os valores dentro dele. Outras regressões poderiam ser formadas, ou até mesmo teste de SVC (Support Vector Classification), para compreender melhor o que está ocorrendo.

**Conclusão**

O teste realizado mostra valores aproximados de probabilidade entre o dataframe utilizado para gerar o código e o outro para o teste. Assim, analisando a tabela de frequência de cada nota, por tom, podemos perceber que ambas estão com resultados muito semelhantes. Assim, o código gerado, manteve-se fiel às deduções e análises feitas.