Relatório de Análise do Classificador para futuro Aplicativo de Previsão de Aceitação de filmes

Projeto 3 Ciências dos Dados

Phelipe Müller
Sabrina Machado
Bruno Cury
SALA - B

Motivação

Se você se considera um entusiasta do âmbito cinematográfico como nós, provavelmente já se encontrou na situação de querer saber o quão bom um filme é (ou vai ser) e não querer assistir o respectivo trailer, devido à grande quantidade de *spoilers* contidos no mesmo.

Partimos daí então com a motivação de desenvolver um mecanismo que, facilitasse o acesso, utilização e visualização de informações confiáveis sobre cinema; estas, completamente imparciais e baseadas inteiramente em análise de dados brutos.

Objetivo

Nosso objetivo é criar uma ferramenta que consiga quantificar com precisão a qualidade de um filme, baseado em informações fornecidas por um *dataset*. Essas informações, depois de convertidas para o formato correto devem ser manipuladas com maestria por um algoritmo *Naive Bayes* Multinomial, que por sua vez retornará (depois de ser propriamente treinado) a nota quantificada do filme em questão.

Ferramentas

API do TMDB

Utilizamos a API do The Movie Data Base para conseguir informaçoes sempre atualizadas dos filmes que já foram lançados para treinar e testar nosso Classificador Naive-Bayes e posteriormente para conseguir informações de filmes que ainda não foram lançados.

Naive-Bayes

Esta é a ferramenta foco do projeto, é possivel separá em dois momentos, o Treino e a Previsão.

Durante o Treino, o nosso classificador recebe as informações de Popularidade, Atores Principais, Diretor e Principais Palavras Chaves, de varios filmes, cada filme em um vetor acompanhado de um inteiro indicando a nota daquele filme. O Classificador, quebra o vetor de entrada, isolando cada um dos termos nele, e adiciona um ao contador associado aquela nota, e repete esse processo para todos os filmes do banco de treinamento.

Após de treinado, podemos realizar um teste, o Classificador preve um numero considerado de filme e nós validamos sua resposta. Para prever a nota de um filme o Classificador recebe um vetor similar aos vetores do Treino, mas não recebe a nota. O Classificador busca em seu banco de dados,

que criou durante o Treino os termos deste vetor, individualmente e calcula a probabilidade do vetor ter cada uma das possiveis notas, baseado no teorema de Bayes (1).

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (1)$$

Depois de calcular cada uma das probabilidades o Naive-Bayes retorna a que for maior entre elas.

Encoder

Para podermos utilizar a biblioteca do Scikit Learn de Naive-Bayes, precisamos que os parâmetros e as varaiveis entrem no Classificador como inteiros e para isso desenvolvemos o que chamos de Encoder.

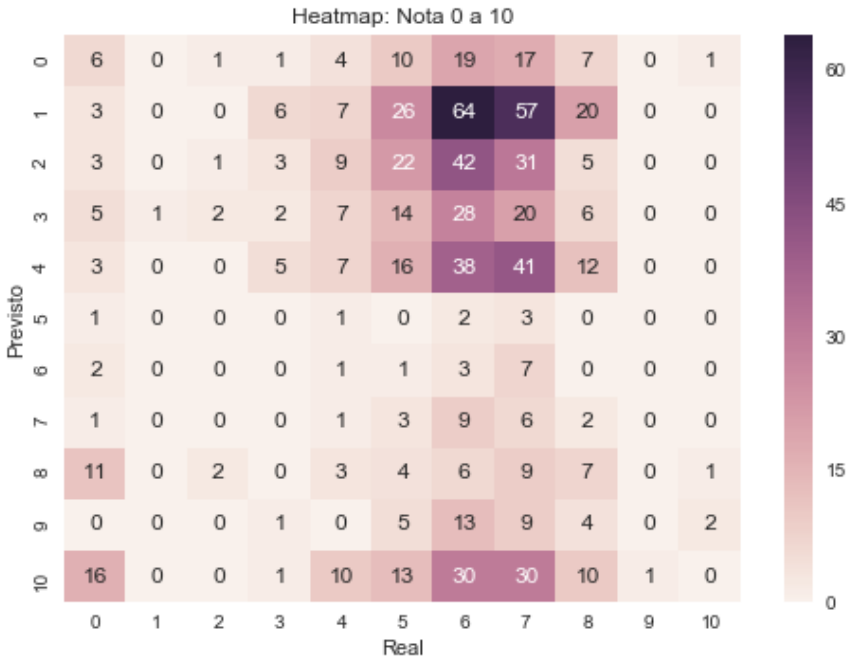
Dentro do Encoder existe uma lista chamada Codex, que recebe Nomes de Atores, Palavras chaves e os associa cada um deles a um inteiro, que chamamos de Index. Por exemplo, ao entrarmos o nome “Adam Sandler” no Encoder, ele pesquisará em seu Codex qual o Index que está associado a Adam Sandler e nos retornará ele, supomos, 32.

Depois que conseguimos tornar nossos atores em numeros, podemos criar o vetor para entrar no Naive-Bayes.

Desenvolvimento

Inicialmente foi usado o *DataFrame* com 5000 filmes do *The Movie Data Base* (TMDB) em conjunto com seu API para obter as informações necessárias para prever sua nota, que consideramos ser a aceitação do público pelo filme (Atores Principais, Diretor, Popularidade e Palavras Chaves do filme). Em seguida foram codificadas as informações de cada filme com o *Encoder,* gerando um vetor de informações para cada filme.

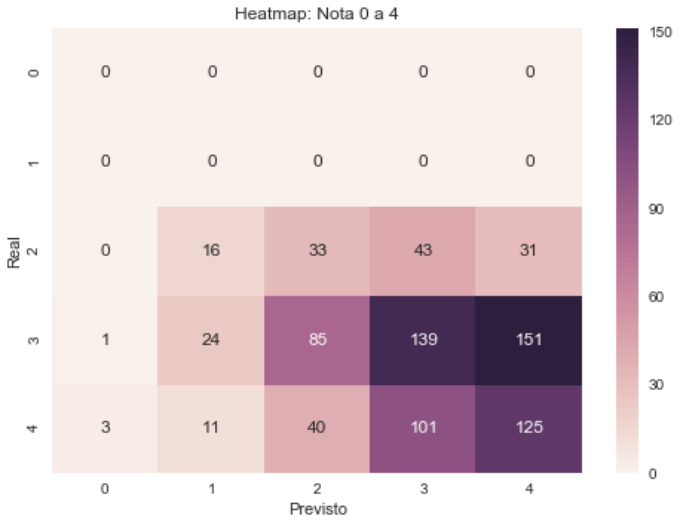
Foi utilizado cerca de 80% dos dados, para treinar o Classificador *Naive-Bayes Multinomial* (da Biblioteca *Scikit Learn*) e outros 20% para realizar o teste e obtendo o seguinte resultado:



Através de um *heatmap*, ou Matriz de confusão, foi possível observar o desempenho do classificador. Dividindo o *heatmap* em 4 setores, dois a cima do centro e dois a baixo, sabe-se que quanto maior for a concentração no setor esquerdo superior e direito inferior mais acertos o classificador está obtendo e quanto mais próximos a diagonal que passa por esses mesmos setores mais precisos estão estes acertos.

Com essa análise, foi preciso tomar medidas para melhorar este resultado, foi optado de ao invés utilizar as notas 0 a 10, utilizar 0 a 4 estrelas, assim minimizando a quantidade de dados necessárias para treinar o Classificador.

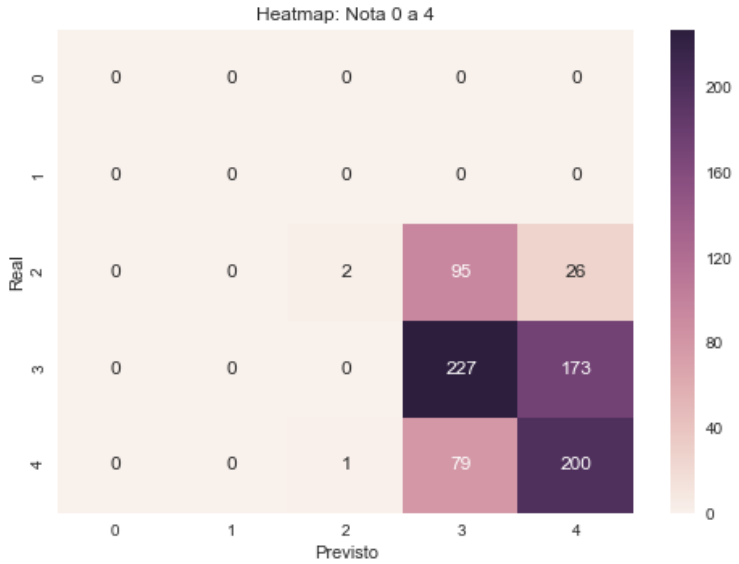
Realizando o mesmo método de análise foi obtido este resultado:



Acerto Absoluto	297(36%)
Acerto Parcial	396(48%)
Erros	126 (15%)

Já bem mais interessante que antes desta iteração. Ocasionalmente não passou quase nenhum filme de baixa qualidade na banda de testes, o que deixa uma lacuna de dúvida sobre seu desempenho ao analisar filmes com esta natureza. Mas relevando este fato, foi obtido um resultado bastante interessante, com pouco erros graves em suas classificações.

Como uma última verificação, foi comparado o desempenho deste Classificador *Naive-Bayes* com um Classificador SVM (*Support Vector Machine*), e realizamos com ele o mesmo método de análise, de Matriz de confusão, que as duas analises a cima, obtendo:



Acerto Absoluto	429(52%)
Acerto Parcial	349(42%)
Erro	27(3%)

Por sua vez o SVM teve uma precisão maior, tornando-o o foco de uma nova analise com o intuito de validar sua qualidade.

Com o intuito de gerar uma aplicação, fizemos uma interface em linha para encontrar um filme desejado e fazer a previsão de sua nota, ainda na versão de desenvolvedor ele compara com a nota real, se ela já existe.

Conclusão

O Classificador *SVM* obteve um desempenho bastante agradável e superior ao Classificador inicial *Naive-Bayes*, viabilizando a continuação do projeto do aplicativo que usará um sistema parecido com o da função" UI", desenhado no Arquivo Principal da análise do projeto.

O aplicativo contara com notas e informações gerais obtidas através do Banco de Dados do *The Movie Data Base*, além de alguns comentários dos usuários do aplicativo, ele usará também as previsões antes do filme sair e a nota real deles após eles saírem para atualizar e calibrar sua performance.

Outras alterações e iterações inclui gerar perfis de usuário e classificar cada um de nossos usuários dentre os perfis possuindo notas diferentes para um mesmo filme. Indicação de filmes dado os filmes pesquisados. E melhorar a performance utilizando os filmes semelhantes ao filme analisado.

Agradecimentos Especiais

Gostaríamos de agradecer aos professores Raul Ikeda e Fabio Ayres por todo suporte e apoio durante o desenvolvimento do projeto.

Agradecemos também aos desenvolvedores do *Scikit Learn* e do *The Movie Data Base*, por desenvolverem e disponibilizarem ferramentas tão essenciais para este projeto.