

Relatório da análise de dados da Google Play Store

Mateus P. Genaro, genaro.mateus@gmail.com

20 de fevereiro de 2020

Sumário

1. Introdução
2. Metodologia
 - a. Pré-processamento
3. Resultados
 - a. Base 1
 - i. Modelo de Naive Bayes
 - ii. Modelo de Árvore de Decisão
 - iii. Modelo de Florestas Aleatórias
 - b. Base 2
 - i. Modelos
 - ii. Palavras-chave
4. Conclusão

1. Introdução

Fundamentando a investigação nas bases de dados da Google Play Store que foram fornecidas, o objetivo principal dessa análise é responder à pergunta “O que faz os usuários gostarem dos aplicativos?”. A resposta mais completa para a indagação se encontra nos comentários dos usuários dos aplicativos. Foi possível chegar a essa conclusão através dos resultados obtidos com os métodos analíticos e estatísticos escolhidos, onde pôde-se ter uma compreensão de quais são as variáveis relevantes que fazem com que os aplicativos sejam apreciados pelos usuários.

2. Metodologia

Para essa análise, motivado pela familiaridade, praticidade e desempenho, como principal recurso, foi utilizado a linguagem de programação Python, juntamente às suas bibliotecas relevantes (por exemplo, pandas, sklearn, numpy, entre outros), e os softwares auxiliares Anaconda e Spyder. Utilizou-se estes tanto na parte de pré-processamento e limpeza da base de dados quanto no desenvolvimento das modelagens usadas na aprendizagem de máquina. Complementarmente, devido à algumas dificuldades técnicas específicas em Python, também foi utilizado o software Microsoft Excel para o pré-processamento. Além disso, para o desenvolvimento de gráficos e dashboard, utilizou-se a ferramenta Tableau, pela

familiaridade e eficiência no desenvolvimento de visualizações de dados. WebGraphviz foi uma ferramenta complementar para gerar a visualização de árvore de decisão.

a. Pré-processamento

Na ‘Base 1’, a qual é a base de dados contendo as informações referentes aos aplicativos e nota média que os usuários deram, através de uma análise qualitativa dos dados, foi escolhido como atributo principal para investigação a nota média. A motivação dessa escolha está na tentativa de responder à pergunta central da análise, pelo fato lógico de que se um usuário avalia um aplicativo com uma nota acima da média (ou seja, com notas de 0 à 5, avalia com uma nota acima de 2,5), espera-se que o usuário tenha apreciado o aplicativo de alguma maneira.

Sendo assim, para a começar a limpeza da base de dados, fez-se uma busca por dados inconsistentes envolvendo as notas médias. Dos 10.840 valores, apenas um único destes foi encontrado como inconsistente. A escolha foi desconsiderar o dado inconsistente para cálculos futuros para não enviesar a análise, além de sua representação quantitativa ínfima comparado ao total de valores. Já pela busca por dados faltantes, foram encontrados 1.474 valores, os quais foram preenchidos com as técnicas de ‘preenchimento retrógrado’, preenchendo o valor faltante com o valor da linha anterior, dentro do mesmo atributo nessa base de dados. Demais técnicas de preenchimento foram utilizadas (como ‘preenchimento adiante’ e média), mas mostraram não alterar os resultados finais. O mesmo método de preenchimento de valores faltantes foi feito para o atributo “Size”. Para o mesmo atributo, devido a dificuldades técnicas no uso da linguagem Python, foi necessário criar uma cópia do banco de dados (o qual o mesmo foi utilizado para fins de cálculo) para se redimensionar os valores para megabytes, além de que, para valores de ‘Varies with device’, substituímos por ‘0’, para fins de cálculo e não enviesamento dos dados com pesos diferentes. Feito isso, na coluna ‘Installs’ foi retirado caracteres especiais para transformar os valores em números inteiros, para fins de cálculo. Ainda, foram feitas transformações de variáveis categóricas para variáveis numéricas das colunas de ‘Category’, ‘Content Rating’ e ‘Genres’ usando ‘label encoder’. Como veremos nos resultados, as técnicas de ‘one hot encoding’ e escalonamento se mostraram ser ineficientes e prejudicar a acurácia dos modelos usados na aprendizagem de máquina.

Para a ‘Base 2’, temos uma base de dados que contém: comentários dos usuários em relação aos aplicativos e a análise de sentimento (a qual já feita por outra equipe do Google); o sentimento do comentário (positivo, neutro e negativo); a polaridade do sentimento; e a subjetividade do sentimento. Após a busca por dados inconsistentes nas colunas ‘Sentiment’, ‘Sentiment Polarity’ e ‘Sentiment Subjectivity’, não foi encontrado nenhum valor inconsistente. Já para valores faltantes, encontramos um valor de 26.863, do total de 64295, ou seja, 41,78% de valores faltantes. Como são valores faltantes para os atributos de ‘Sentiment’, ‘Sentiment Polarity’ e ‘Sentiment Subjectivity’ simultaneamente, tendo apenas a informação da coluna ‘App’ preenchida, então, a decisão foi desconsiderar todas as linhas faltantes para a criação de um ‘dataframe’, uma vez que, a priori, não há uma maneira eficaz de se preencher os dados faltantes de modo a se obter valores não enviesados e consistentes com a base de dados. Além disso, fez-se o

'label encoding' do atributo 'Sentiment', e foi observado não ser necessário o escalonamento e padronização do 'dataframe', uma vez que 'Sentiment Polarity' e 'Sentiment Subjectivity' já se encontram com valores normalizados.

3. Resultados

a. Base 1

Em primeira instância, observa-se que ambas bases de dados tem diversas variáveis categóricas. Como dito anteriormente, para a Base 1, o atributo que foi classificado como mais relevante foi 'Rating' (nota média por aplicativo), e, desta forma, gostaríamos de que, utilizando aprendizagem de máquina, consiga-se encontrar quais são os outros atributos, nesta mesma base de dados, que tem um valor de correlação relevante, ou seja, quais as demais variáveis mais influenciam para determinar a nota média. Então, uma abordagem válida para se estimar as intensidades dessas correlações é usar os atributos relevantes melhor consigam prever o valor da nota, dados as demais informações sobre os mesmos atributos.

Antes de aplicarmos os modelos de aprendizagem de máquina, pôde-se verificar algumas informações relevantes dessa base:

Média total das notas médias por aplicativo: 3.40

Desvio padrão das notas médias: 1.09

Nota mínima: 1.00

Nota máxima: 5.00

Top 3 categorias com as maiores notas médias agrupadas por categoria:

Education (4.37); Art and Design (4.20); e Game (4.15)

Top 3 categorias com as menores notas médias agrupadas por categoria:

Business (2.80); Medical (3.09); e Libraries and Demo (3.13)

A seguir, temos o gráfico que mostra as notas médias, agrupadas por categoria, para cada categoria, plotado como 'Category' por 'Rating', onde 'Average' é a média total:

Gráfico de Category por Rating

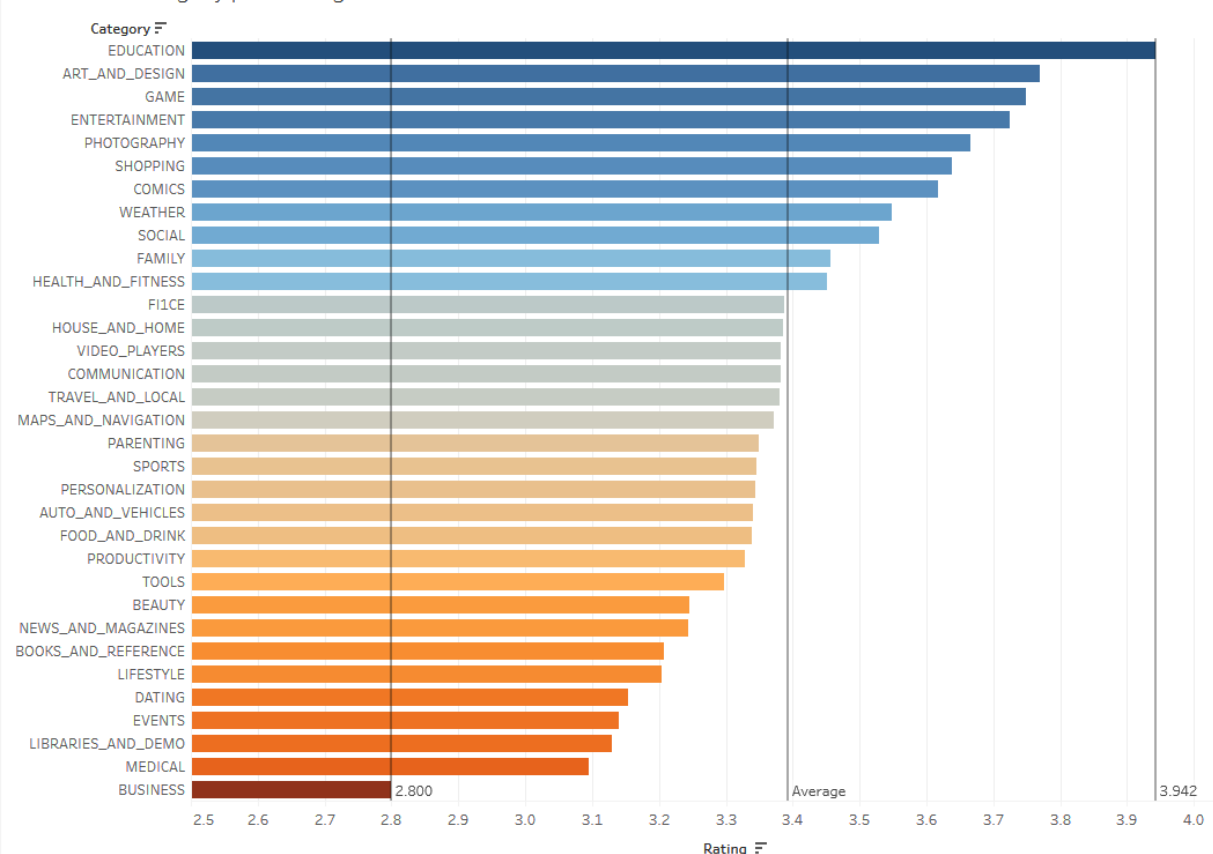


Fig.1: Gráfico que mostra as notas médias, agrupadas por categoria, para cada categoria

A partir disso, podemos ver que se poderia direcionar uma coleta de dados mais detalhada voltada para as categorias dos aplicativos com maiores e menores notas médias, para se obter novas variáveis que ajudem a determinar o que fazem essas categorias serem bem avaliadas, ou não.

Partindo para a parte de aprendizagem de máquina, foram escolhidos três tipos de modelagem que melhor se enquadram para essa situação de aprendizagem não supervisionada, com variáveis categóricas e que, para fins práticos, tenha um nível de complexidade médio. São esses: modelo de Naive Bayes; modelo de Árvore de Decisão; e modelo de Florestas Aleatórias.

i. Modelo de Naive Bayes

O resultado com maior acurácia foi com um pré-processamento de ‘label encoder’ e escalonamento, com um valor de **precisão = 19,43%**.

Obteve-se a seguinte matriz de confusão:



Fig.2: Gráfico ‘heat map’ para a matriz de confusão no modelo de Naive Bayes.

Para este modelo, conclui-se que se mostra completamente insuficiente e ineficiente para prever Rating, dados os demais atributos.

ii. Modelo de Árvore de Decisão

O resultado com maior acurácia e menor complexidade foi usando o pré-processamento de ‘label encoder’ e escolhendo um número ‘n’ de ‘folhas’ na árvore com $n=3$. Com isso, obteve-se um valor de **precisão = 74,05%**, o que já se nota que é uma melhoria bem considerável.

Desta forma, foi gerada a seguinte matriz de confusão:

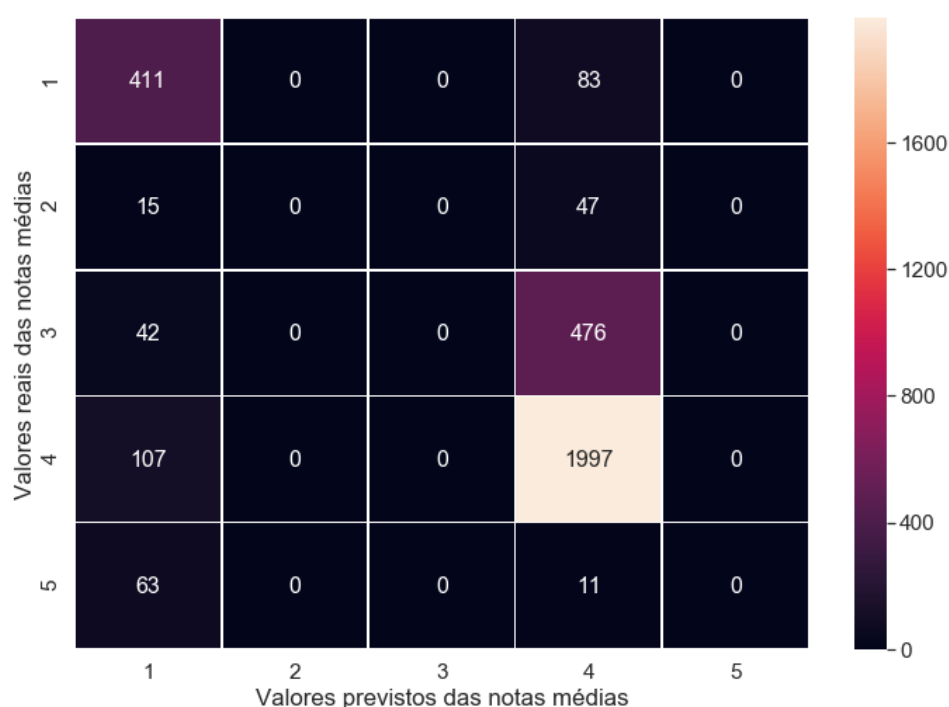


Fig.3: Gráfico ‘heat map’ para a matriz de confusão no modelo de Árvore de Decisão com 3 nodos.

Podemos observar então que este modelo de Árvore de Decisão consegue prever com uma precisão razoável, apesar de ser significativamente melhor que o modelo de Naive Bayes. Para estudar melhor quais são as variáveis que melhor consegue prever as notas médias, usamos uma visualização gráfica do modelo:

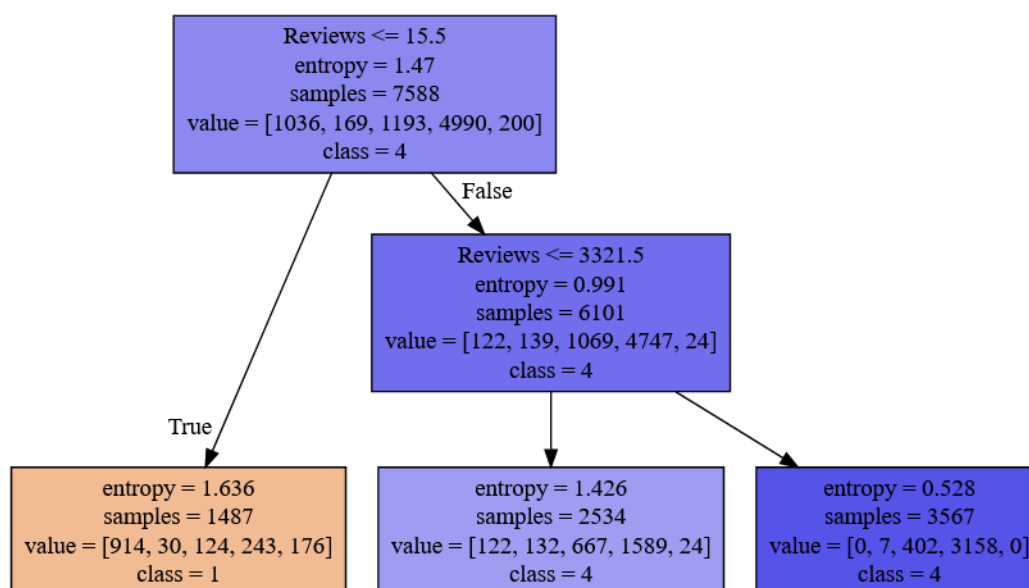


Fig.4: Gráfico de Árvore de Decisão com 3 nodos para a Base 1.

Usando a entropia como parâmetro de informação, podemos concluir que a melhor maneira de se classificar as notas médias para cada aplicativo é utilizando o número total de avaliações que aquele aplicativo tem. O que isso realmente quer dizer é que, usuários tem uma tendência maior de avaliar um aplicativo se ele é, de alguma forma, bem apreciado ou desagradável, visto que os valores que foram em maior parte previstos corretamente são os com notas 4 e 1, o que uma análise qualitativa dos dados pode reafirmar esse resultado.

Ainda, obteve-se um resultado para a precisão com um número livre de n folhas, mas mostrou-se ter um valor menor do que para com $n=3$ (68,29%), e um aumento exponencial de complexidade, fazendo um ‘overfitting’ dos dados. Para um número n de folhas com $n=10$, tivemos uma melhora de 0,74% na precisão, mas com um número significativo de aumento de complexidade, o que, pela quantidade de acréscimo na precisão, não é uma situação relevante, neste caso. Além disso, para $n=10$, o atributo ‘Installs’ mostrou-se ser a segunda variável mais relevante para o modelo, o que faz sentido, uma vez que um aplicativo que é bem apreciado e possivelmente útil, faz com que mais usuários o instalem em seus aparelhos eletrônicos.

iii. Modelo de Florestas Aleatórias

Dado o resultado do modelo de Árvore de Decisão, decidiu-se fazer a extensão do seu modelo para Florestas Aleatórias, afim de obter um resultado mais preciso. Sendo assim, o resultado com um pré-processamento de ‘label encoder’ e escolhendo um número ‘ k ’ de ‘árvores’ na floresta com $k=400$, obteve um valor de **precisão = 74,35%**. Ou seja, mesmo para um k relativamente grande, se foi capaz de aprimorar a precisão em apenas 0,3%, o que é interessante, mostrando que realmente o atributo ‘Reviews’ desempenha um papel importante para se prever ‘Rating’.

A matriz de confusão para este modelo é dada da seguinte forma:

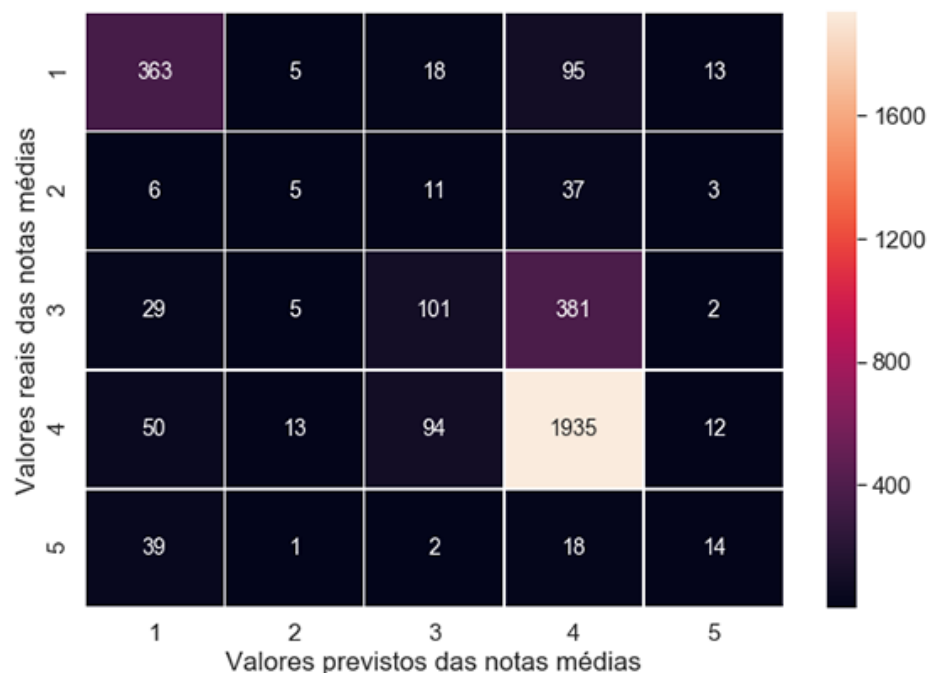


Fig.5: Gráfico ‘heat map’ para a matriz de confusão no modelo de Florestas Aleatórias com $k=400$ para a Base 1.

Vemos que nessa matriz de confusão temos uma dispersão maior de valores entre as notas médias reais e previstas, o que indica que é um modelo mais robusto e com uma performance melhor na tentativa de predição dos valores. Ainda, foi também calculado a acurácia para um número menor de k . Para $k=40$, obteve-se um valor de precisão = 73,80%, que mostra um desempenho pior do que um modelo, relativamente, simples como o de árvore de decisão com apenas 3 nodos.

Ademais, afim de melhorar os nossos resultados, dentro desse modelo, usamos dois métodos de hiper parametrização, o chamado ‘tunning’, que é uma aprimoração do próprio modelo. Escolhendo os valores dos parâmetros que vão maximizar a precisão do modelo, foi possível obter uma **precisão = 75,52%**. Resultado esse que mostra uma melhora com um acréscimo de 1,17% na precisão. Apesar de que em alguns casos 1% de aumento de precisão podem gerar algum tipo de lucro, dentro do contexto dos atributos sendo analisados, este não é um resultado que contribui de maneira significativa.

De maneira complementar, foi feita uma investigação da Base 1 usando o modelo de K-means para clusterização, mas não mostrou nenhum resultado relevante.

b. Base 2

Como a Base 1 mostrou uma precisão apenas razoável, por volta de 74%, e com apenas um atributo relevante para tentar predizer as boas avaliações dos aplicativos, vamos

também investigar a Base 2 na busca de achar uma resposta mais interessante e acurada para a pergunta central da análise. Nota-se, em primeira instância, uma quantidade menor de atributos e de variáveis categóricas, o que não necessariamente significa que teremos resultados melhores ou piores que na Base 1, a priori. Em uma análise qualitativa, o atributo com maior relevância e significado é ‘Sentiment’, que são variáveis categóricas geradas a partir da análise de sentimento, já feita por outra equipe do Google, a qual o mesmo será usado como o conjunto de dados que gostaríamos de prever dado os valores dos atributos de ‘Sentiment Polarity’ e ‘Sentiment Subjectivity’.

Assim como anteriormente, antes de abordarmos o problema com os modelos de aprendizagem de máquina, pôde-se verificar algumas informações dessa base, a partir dos atributos ‘Sentiment Polarity’ e ‘Sentiment Subjectivity’:

Parâmetros	Sentiment Polarity	Sentiment Subjectivity
Média	0.18	0.49
Desv. P.	0.35	0.26
Máximo	1	1
Mínimo	-1	0
Positivo	0.37	0.57
Neutro	0	0.08
Negativo	-0.26	0.53

Fig. 6: Tabela com valores dos parâmetros estatísticos por polaridade e subjetividade de sentimentos

Onde ‘Desv. P.’ é o desvio padrão, ‘Positivo’, ‘Neutro’ e ‘Negativo’ são as médias de cada variável agrupadas pelos respectivos seus sentimentos. A princípio, olhando somente para esses dados, não se é capaz de inferir alguma informação relevante através de uma análise qualitativa. O que se pode dizer é que, em média, os comentários que foram julgados como sentimentos neutros tem o menor valor de subjetividade sentimental. Também se obteve um número de 23.998 comentários classificados como de sentimento positivo, 8.271 como negativo e 5.158 neutro, somando um total de 37.427 valores de ‘Translated Review’.

Sendo assim, utilizaremos os modelos de aprendizagem de máquina para se fazer uma análise quantitativa mais profunda dos dados. Da mesma maneira como na Base 1, e motivados pelas informações, foram escolhidos os mesmos três tipos de modelagem: Naive Bayes, Árvore de Decisão e modelo de Florestas Aleatórias.

i. Modelos

Para todos os modelos utilizados foram utilizados os mesmos pré-processamentos de ‘label encoder’, pelo fato de resultarem em uma maior precisão. Segue os resultados das precisões para cada modelo:

- Naive Bayes = **96,94%**;
- Árvore de decisão (n = 3) = **99,88%**;

- Florestas Aleatórias (n = 400) = **99,86%**.

Iminentemente, já percebemos a diferença e a considerável precisão desses resultados de precisão quando comparados a Base 1. Esses resultados mostram que a base é auto consistente ao usar os atributos de ‘Sentiment Polarity’ e ‘Sentiment Subjetivity’ para classificar o tipo de sentimento com que os valores estão ligados, o que mostra que a equipe que fez a análise prévia fez um ótimo trabalho.

Com isso, obtivemos dois resultados interessantes da aplicação desses modelos. O primeiro é o gráfico da árvore de decisão com n = 3:

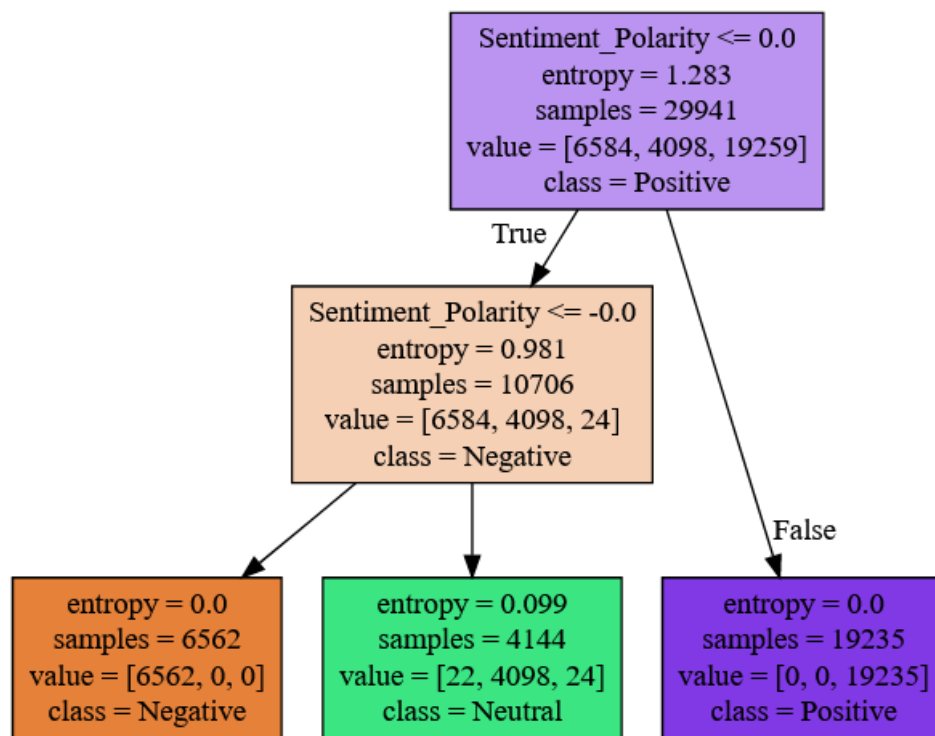


Fig.7: Gráfico de Árvore de Decisão com 3 nodos para a Base 2.

E o segundo, é a sua respectiva matriz de confusão:

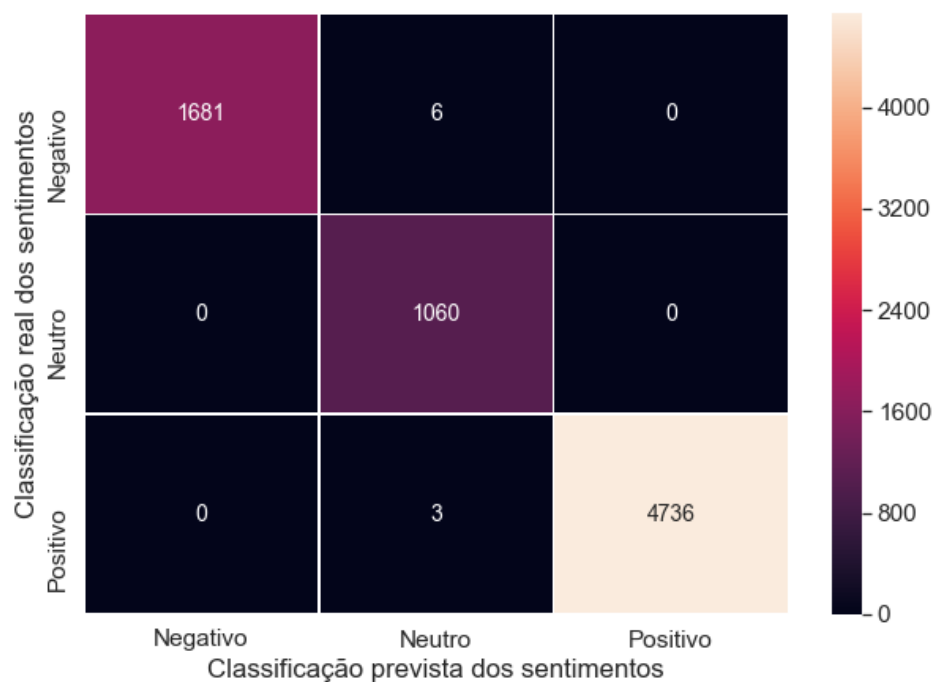


Fig. 8: Gráfico ‘heat map’ para a matriz de confusão no modelo de Florestas Aleatórias com $k=400$ para Base 2.

Na **Fig. 7**, é fácil perceber que ‘Sentiment Polarity’ desempenha um papel muito mais relevante que ‘Sentiment Subjectivity’ para realizar a classificação dos sentimentos, o que é um resultado esperado, porém não com tanta dominância. E na **Fig. 8**, podemos notar que o modelo erra apenas 9 previsões de resultados, dos 7.486.

Assim, afim de investigar mais a fundo a relação de polaridade e subjetividade de sentimentos e como são suas classificações, também aplicamos um modelo de K-Means, o qual consiste em fazer agrupamentos de dados em grupos que os quais dados tem possíveis características semelhantes, o chamado ‘clustering’. Feito o escolamento dos dados, obteve-se o seguinte gráfico de ‘Sentiment Subjectivity’ em função de ‘Sentiment Polarity’:

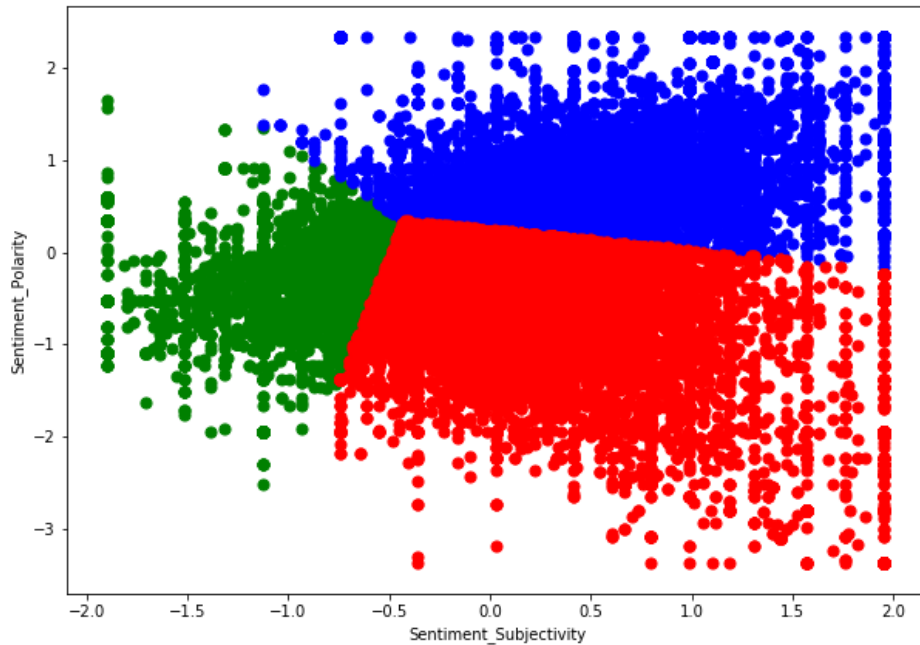


Fig. 9: Gráfico de dispersão com 3 clusters dos dados escalonados de ‘Sentiment Subjectivity’ em função de ‘Sentiment Polarity’, para a Base 2

O gráfico acima reforça a ideia de classificação em 3 tipos de sentimentos, através dos atributos de ‘Sentiment Subjectivity’ e ‘Sentiment Polarity’. Como esperado, vemos que há uma maior densidade de pontos para o sentimento neutro concentrada entre os valores entre -1 e 1 (lembrando que reescalamos os valores para o plot) para polaridade e também tem baixo valor de subjetividade, que é o que se vê no banco de dados. O mesmo acontece para os sentimentos positivo e negativo, com valores de densidades maiores de $0,5$ à 2 e de 0 à -2 , respectivamente, onde há uma clara de divisão, a partir de $-0,5$ de subjetividade, bem próxima ao valor zero em polaridade. O que é interessante é quando, ao invés de usar 3 clusters, escolhemos um valor considerado ótimo de 5 clusters:

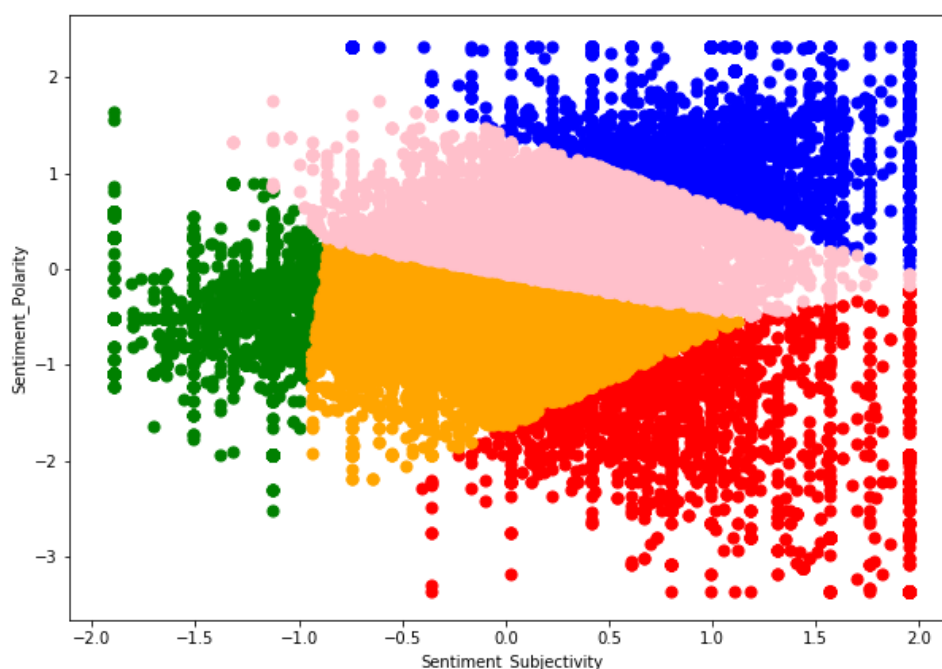


Fig. 10: Gráfico de dispersão com 5 clusters dos dados escalonados de ‘Sentiment Subjectivity’ em função de ‘Sentiment Polarity’, para a Base 2

Percebemos agora que as linhas que separavam os clusters com $n=3$, anteriormente, sofreram um deslocamento e novas duas linhas de separação apareceram. Os dois novos clusters que são mostrados, de cores rosa e laranja, tem altas densidades de pontos mais centralizadas próximos ao valor zero de cada atributo, enquanto os clusters azul e vermelho foram reclassificados como os de alto valor de polaridade e baixo valor de polaridade, respectivamente. À vista disso, somando uma análise qualitativa de observação dos dados em ‘Translated Review’ com essas novas informações, podemos concluir que duas novas categorias de sentimento podem ser classificadas, neste caso. O cluster rosa pode ser classificado como o novo sentimento ‘pouco positivo’, enquanto o azul seria classificado, possivelmente, como ‘muito positivo’. O mesmo é válido para o sentimento negativo, com as cores laranja sendo ‘pouco negativo’ e vermelho ‘muito negativo’. Essa reclassificação seria interessante se executada oficialmente pela equipe especializada, pois, desta maneira, conseguiríamos ainda mais detalhes da base de dados e, conseqüentemente, podemos fazer uma análise mais assertiva, assim como extrair informações mais específicas para futuras investigações.

Apesar de mostrar a auto consistência da base de dados da Base 2, através das próprias precisões, e nos apresentar a importância do atributo de polaridade de sentimento, esses resultados ainda não fazem com que respondamos de maneira satisfatória a pergunta principal desta análise como um todo: “O que faz os usuários gostarem dos aplicativos?”. Como acabamos de provar a coesão e coerência dos dados para a classificação dos sentimentos, sendo que os mesmos foram estimados fazendo uma análise a partir dos comentários, então, devemos ir mais a fundo e usar os próprios comentários do atributo ‘Translated Review’ para se obter uma resposta que seja mais relevante para a solução desse problema.

ii. Palavras-chave

Com base nos resultados obtidos até o momento, fez-se uma avaliação estatística do atributo 'Translated Review' usando o parâmetro de palavras mais comuns que aparecem nos comentários. Foram desconsideradas as palavras de parada (conjunto de palavras mais usadas em qualquer idioma, por exemplo, em inglês, "i", "you", "the", "is", "and", entre outras), afim de ter somente, o que é foi definido como, palavras-chave, as quais são palavras que realmente sejam significantes e mostrem algum indicativo de tipo de sentimento (positivo, negativo ou neutro). Então, através de uma análise qualitativa, foram escolhidas palavras-chave que tem maior representatividade e significado para dar um porquê de um usuário apreciar ou depreciar determinado aplicativo que está avaliando.

Selecionando as 100 palavras mais comuns, obteve-se uma quantidade de total de 219.694, das quais, 148.449 estão classificadas dentro da categoria de sentimento positivo, 56.746 dentro do sentimento negativo, e 14.499 neutro. Passando o filtro de palavras de parada e escolhendo 20 palavras-chave apropriadas, obtivemos um total de 48.538 quantidade total de palavras significante que aparecem nos comentários. Desse montante, 31.637 são relacionadas com sentimento positivo, 13.758 ao sentimento negativo e 3.143 ao sentimento neutro.

Tendo esses resultados interessantes, foram gerados 3 tipos de gráficos para representar esses resultados:



Fig. 11: Gráfico de nuvem de palavras contendo as palavras mais comuns em 'Translated Review' agrupadas por sentimentos.

Na figura acima pode-se observar um gráfico de nuvem de palavras (o qual o tamanho das fontes são diretamente proporcionais com a quantidade de palavras), as quais as palavras que aparecem aqui são as palavras-chave escolhidas, como já explanado. Pode-se notar então que palavras como “game” podem tanto estar associadas com aplicativos de jogos que, possivelmente, tem um bom desempenho e entretenimento, quanto com jogos com má performance e entediante. Enquanto isso, podemos notar outras palavras que não são tão ambíguas e tem uma maior polarização e menor subjetividade, como, para palavras positivas “time”, “easy”, “free”, love”, etc, e para neutras e negativas palavras como “money”, “work”, “update”, “problem”.

Na **Fig. 12**, a seguir, o que se nota de interessante é a dominância das palavras positivas em relação as palavras negativas, representando 65,18% do total, enquanto as palavras negativas tem um percentual de 28,34% do total e as neutras 6,47%.

Apesar das palavras positivas representarem uma maioria, podemos ver que ainda há bastante espaço para aumentar esse número, aprimorando os aplicativos de tal maneira a se basear nos resultados dessa mesma análise.

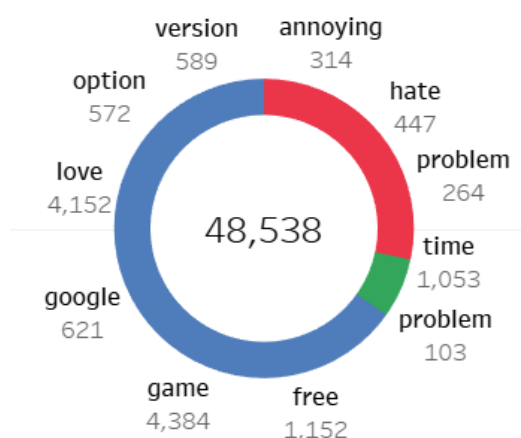


Fig. 12: Gráfico de rosquinha mostrando as somas das quantidades de palavras mais comuns em ‘Translated Review’ agrupadas por sentimentos, com algumas palavras-chave dentro dos grupos e suas respectivas quantidades. No centro do gráfico, temos a soma do total de palavras-chave.

Como último resultado dessa análise, temos o gráfico a seguir:

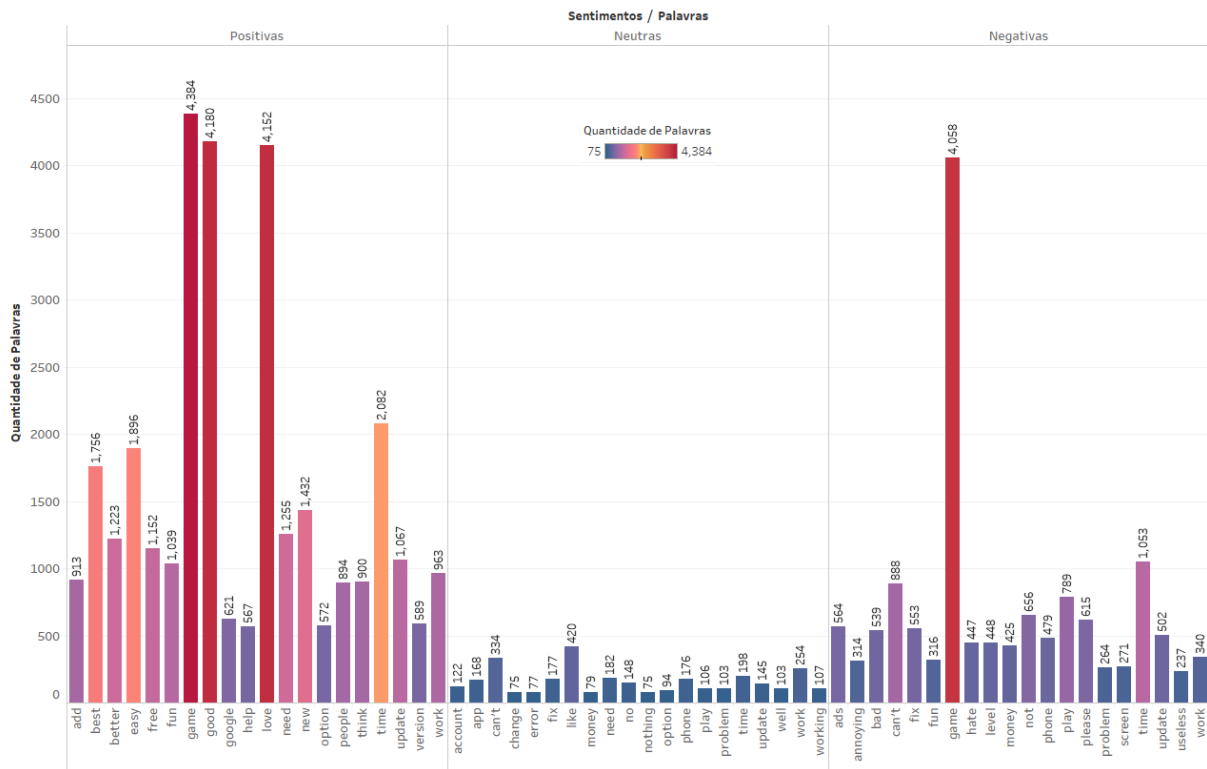


Fig. 13: Gráfico de barras com as respectivas quantidades de cada palavra-chave, agrupadas pelas suas classificações de sentimento.

Observando a **Fig. 13**, podemos notar mais facilmente a predominância de algumas palavras-chave sobre outras, como no caso do ‘outlier’ da palavra ‘game’ no grupo das palavras negativas. Interpretando este resultado específico, nota-se que este é realmente uma variável que influencia fortemente na apreciação do aplicativo pelo o usuário e deveria ser considerada durante o desenvolvimento e nos futuros reparos para os aplicativos de jogos classificados como negativos e/ou com notas baixas.

4. Conclusão

O conjunto das devidas análises realizadas em cada uma das duas bases de dados nos proporciona resultados que, quando combinados, nos fornece uma possível resposta para a pergunta primordial: “O que faz os usuários gostarem dos aplicativos?”. Em uma curta resposta: qualidade. A resposta mais completa, e complexa, se dá com os nossos resultados obtidos. A partir da Base 1, se pôde determinar, através dos parâmetros e modelagens estatísticas usadas, que o atributo ‘Reviews’, ou seja, o número de avaliações que um aplicativo tem, é diretamente proporcional a notas médias altas e suas notas médias baixas. Isto é, o número de avaliações é tão grande para notas altas quanto para notas baixas, o que pode se concluir que a satisfação e insatisfação dos usuários está diretamente relacionada a qualidade geral dos aplicativos. Na base de dados da Base 2, foi possível obter resultados mais satisfatórios, com relação a pergunta, de maneira que, tendo uma base de dados auto consistente que faz uma apurada avaliação e classificação de sentimentos, através dos comentários dos usuários para cada aplicativo, foi possível determinar que usuários tem um

maior apreço por aplicativos que: são jogos com boa qualidade, pela palavra ‘game’; os fazem, de alguma forma, otimizarem o seu tempo investido, pela palavra ‘time’; por ser relativamente fácil de usar, pela palavra ‘easy’; por ser um aplicativo recente, pela palavra ‘new’; por ser divertido, gratuito, funcional, entre outras características, que podem ser observadas nas palavras do gráfico da **Fig. 13**. Somado a isso, também podemos realizar o aprimoramento dos aplicativos fazendo uma análise das palavras relacionadas ao sentimento negativo. Do que pode ser melhorado –ou a ser considerado em futuros desenvolvimentos–, temos: desempenho e entretenimento de jogos, pela palavra ‘game’; não demandar, de alguma maneira, um longo tempo de uso, pela palavra ‘time’; ajustar a quantidade de propagandas, pela palavra ‘ads’; ser mais atualizado, pela palavra ‘update’; menos problemático, pela palavra ‘problem’; e assim por diante, observando o mesmo gráfico como para as palavras positivas.

Para perspectivas futuras, podemos, após uma mineração de dados, utilizar aprendizagem de máquina para aplicar modelos mais sofisticados, como os de linguagem natural e n-gramas, afim de preencher os dados faltantes para a Base 2 com comentários gerados aleatoriamente. Com isso, utilizar modelo de redes neurais e deep learning para fazer uma reavaliação e reclassificação de sentimentos, através dos comentários gerados, e, se possível, aprimorar essa própria classificação de sentimentos em mais categorias (como, por exemplo, Muito Positivo, Pouco Positivo, etc), como já previmos através do modelo de K-Means, para garantirmos e prevermos de uma forma ainda melhor as variáveis que fazem com que os usuários gostem dos aplicativos da Google Play Store.