Introdução

O case apresenta três tarefas referentes ao desafio de capturar os dados da Google Play, as quais foram realizadas separadamente cada uma.

Apresentarei em ordem e explicando o que foi feito em cada tarefa.

Tarefa 01

A IDE utilizada inicialmente para os tratamentos é a PyCharm.

Inicialmente é importada as bibliotecas utilizadas para o trabalho:

```
import pandas as pd
import numpy as np
from google_play_scraper import reviews_all
```

É feito o uso do Pandas, Numpy e o Google_play_scraper.

Após a importação, se faz a requisição dos dados com a biblioteca da Google Play, passando como parâmetros a linguagem, o país e qual o endereço do aplicativo, no caso Alexa, a ser utilizado:

```
alexa_resultado = reviews_all(
    "com.amazon.dee.app",
    lang="pt",
    country="br"
)

alexa_df = pd.DataFrame.from_dict(alexa_resultado)
```

É feita uma verificação de cada coluna individualmente e como no exemplo que foi dado eu removi as colunas que não vão ser utilizadas:

```
alexa_df = alexa_df.drop(["reviewId", "userName", "userImage", "replyContent", "repliedAt"], axis=1)
```

Após isso é feito um tratamento e limpeza dos dados, os seguintes tratamentos foram realizados:

- Palavras e títulos padronizadas com uppercase
- Remoção de espaços em branco nos extremos de palavras
- Remoção de aspas duplas
- Remoção de acentuação
- Remoção de números nulos e nan

```
alexa_df.columns = [x.upper() for x in alexa_df.columns]

alexa_df["CONTENT"] = alexa_df["CONTENT"].str.strip()
alexa_df["CONTENT"] = alexa_df["CONTENT"].str.upper()
alexa_df["CONTENT"] = alexa_df["CONTENT"].replace("", '\"')

cols = alexa_df.select_dtypes(include=[np.object]).columns
alexa_df[cols] = alexa_df[cols].apply(lambda x:
x.str.normalize('NFKD').str.encode('ascii', errors='ignore').str.decode('utf-8'))

alexa_df = alexa_df.dropna()
```

Também foi verificado se todos os valores estão em seu respectivos tipos, e confirmou-se que estão então não foi necessária nenhum tipo de conversão de dados:

```
alexa_df.info()
```

```
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   CONTENT               27390 non-null  object
 1   SCORE                 27390 non-null  int64
 2   THUMBSUPCOUNT         27390 non-null  int64
 3   REVIEWCREATEDVERSION  27390 non-null  object
 4   AT                    27390 non-null  datetime64[ns]
dtypes: datetime64[ns](1), int64(2), object(2)
memory usage: 1.0+ MB
```

Após esse tratamento, é feita então a separação entre valores de score positivos, neutros e negativos.
Como está sempre chegando novos comentários no aplicativo, foi realizado um corte dos 10 últimos comentários feitos, assim não resultará em problemas para o banco mais adiante:

```
positivo = alexa_df.loc[alexa_df["SCORE"] >= 4]
positivo = positivo[:-10]

neutro = alexa_df.loc[alexa_df["SCORE"] == 3]
neutro = neutro[:-10]

negativo = alexa_df.loc[alexa_df["SCORE"] < 3]
negativo = negativo[:-10]
```

E então é feita a exportação de cada DataFrame para três arquivos CSV, que representam a faixa do score de positivo, neutro e negativo:

positivo.to_csv(r"C:/Users/Mayara Lopes/Desktop/sauter/positivo.csv", index=False)
neutro.to_csv(r"C:/Users/Mayara Lopes/Desktop/sauter/neutro.csv", index=False)
negativo.to_csv(r"C:/Users/Mayara Lopes/Desktop/sauter/negativo.csv", index=False)

Com esses três arquivos prontos podemos agora realizar a visualização e análise dos dados através da biblioteca Pandas_profiling, a qual é conveniente utilizar no Jupyter Notebook para uma melhor visualização desses dados.

No Jupyter estou importando apenas as seguintes bibliotecas:

import pandas as pd
from pandas_profiling import ProfileReport

E após carregar os arquivos com o código abaixo eu começo a realizar as visualizações individualmente de cada arquivo:

positivo = pd.read_csv("positivo.csv")
neutro = pd.read_csv("neutro.csv")
negativo = pd.read_csv("negativo.csv")

É criado um objeto de visualização do Pandas Profiling para cada dataset separadamente, começando pelo report dos comentários positivos apenas:

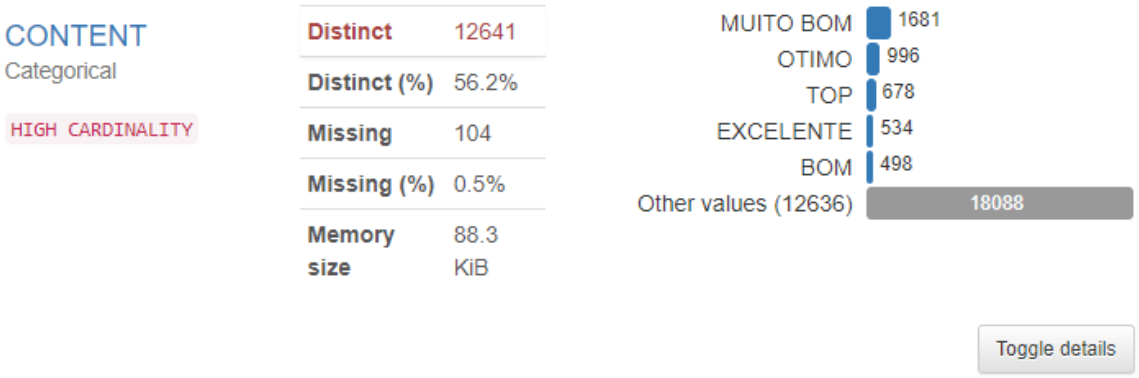
Dataset Positivo

view_positivo = ProfileReport(positivo)

A partir daqui estarei comentando as principais análises feitas e adquiridas com o Pandas Profiling:

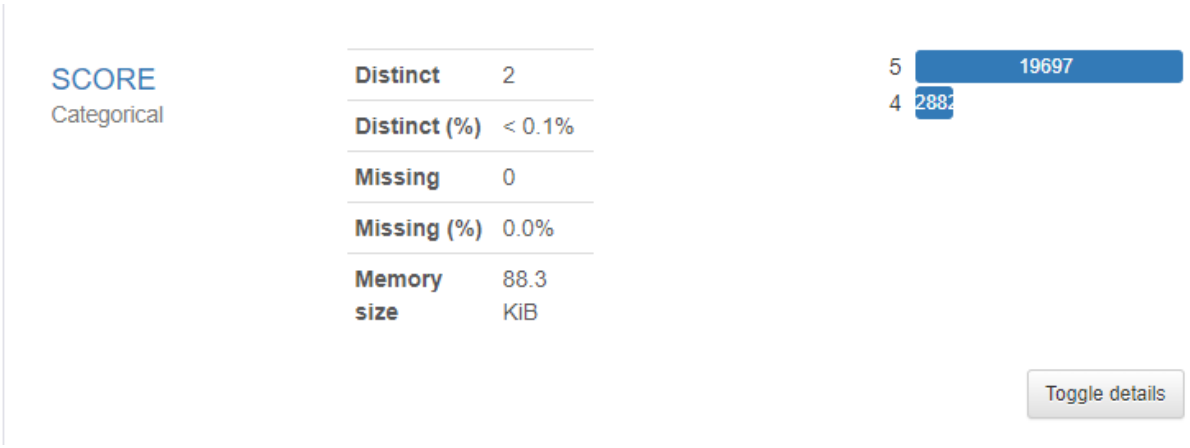
Começamos verificando a aba de alertas, o que nos mostra as características mais atenuantes de cada coluna.
A coluna de CONTENT, REVIWECREATEDVERSION e AT possuem alta cardinalidade pelo fato de terem muitas variáveis categóricas distintas, um fator comum de variáveis categóricas nessa situação.

Se aprofundando na coluna de CONTENT:



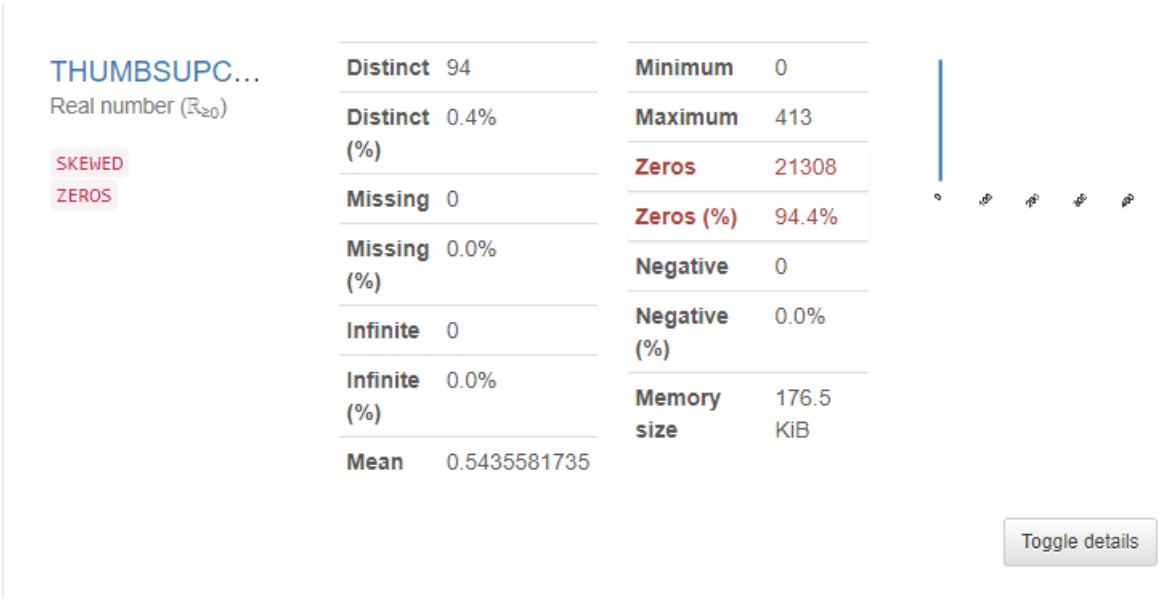
Percebemos que a maioria dos resultados são frases como “Muito bom”, “Ótimo”, “Top”, “Excelente” e “Bom”, o que nos mostra a relação positiva entre os comentários e a pontuação que os usuários deram para o APP.

Na coluna de SCORE:



Observamos a drástica diferença entre os valores de 4 e 5, sendo que o valor 4 representa apenas 14% da base ao todo de positivos. É mostrado também uma possível tendência dos clientes preferirem avaliar como 5 o aplicativo invés de colocarem um valor “menos significante”.

Na coluna THUMBSUP temos:



94 valores diferentes para a quantidade de votos que as avaliações receberam, e nos é mostrado que o mínimo de votos foi 0, enquanto o máximo foi 413 para um único comentário. Como a média é de 0.54 para os votos, podemos perceber que se for plotado um gráfico, a tendência dele é ter uma curva voltada para a esquerda, sendo a maioria dos votos nulos, ou seja, sem ter recebido voto de usuário algum.

Começando a analisar o dataset de valores neutros, podemos verificar os seguintes dados que nos foram mostrados:

Dataset Neutro

view_neutro = ProfileReport(neutro)

Overview | Alerts 10 | Reproduction

Alerts

SCORE has constant value "3"	Constant
CONTENT has a high cardinality: 1416 distinct values	High cardinality
REVIEWCREATEDVERSION has a high cardinality: 77 distinct values	High cardinality
AT has a high cardinality: 1484 distinct values	High cardinality
REVIEWCREATEDVERSION is highly correlated with SCORE	High correlation
SCORE is highly correlated with REVIEWCREATEDVERSION	High correlation
CONTENT is uniformly distributed	Uniform
AT is uniformly distributed	Uniform
AT has unique values	Unique
THUMBSUPCOUNT has 1105 (74.5%) zeros	Zeros

Observamos inicialmente os alertas que foram gerados, e como podemos ver logo o SCORE tem a variável constante de valor 3, o que é fato, já que foram escolhidos apenas valores que fossem neutros de valor 3.

É interessante perceber que SCORE está altamente correlacionado com REVIEWCREATEDVERSION e abaixo estou explicando o possível motivo:

Overview | Categories | Words | Characters

Value	Count	Frequency (%)
2.2.375370.0	128	8.6%
2.2.438005.0	59	4.0%
2.2.416420.0	58	3.9%
2.2.407457.0	53	3.6%
2.2.436689.0	51	3.4%
2.2.403931.0	47	3.2%
2.2.307833.0	38	2.6%
2.2.347119.0	37	2.5%
2.2.410255.0	37	2.5%
2.2.390493.0	36	2.4%
Other values (67)	940	63.3%

Como vemos, há uma boa distribuição entre os dados, o que nos faz pensar que talvez as pessoas que tenham dado negativo seja porque não seja a versão de aplicativo o problema para elas, mas sim talvez a interface ou outros problemas internos mesmo.

Analisando alguns comentários temos os 5 primeiros e os 5 últimos:

0	APP BOM, MAS TEM MUITO PARA MELHORAR. ALGUMAS CONFIGURACOES NAO SAO INTUITIVAS
1	PODERIA SER MAIS RAPIDO O APP
2	O APP E BOM, MAS LENTO, COM TEMPO DE RESPOSTA MUITO DEMORADO. DEVERIA TAMBEM HA
3	BOAS FUNCOES, POREM E MUITO LENTO!
4	FALTOU UMA BARRA DE PESQUISA

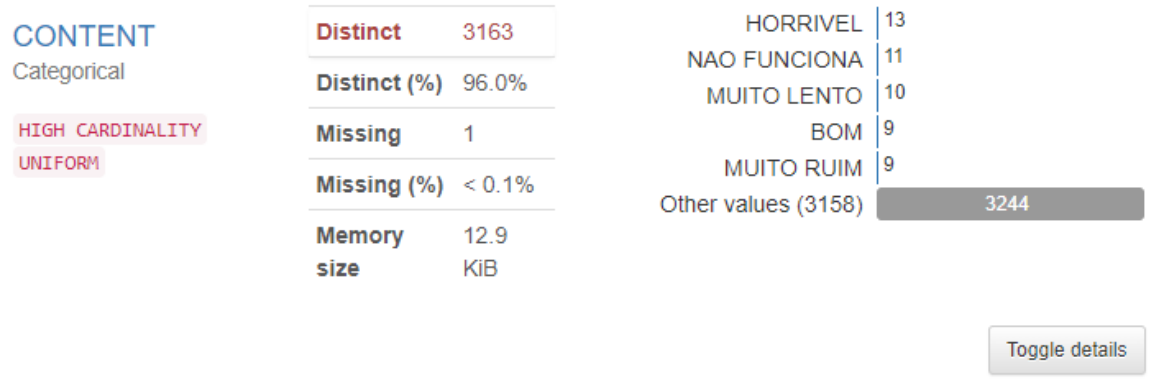
1479	A ALEXA AINDA NAO CONSEGUE RESPONDER MUITA COISA, CREIO QUE EM BREVE A AMAZON
1480	TENHO O PRIMEIRO MODELO DE TORRE DO ECHO (ALEXA) .... GOSTARIA DE SABER SE ELA V
1481	AINDA NAO TEM A OPCAO DE TV E VIDEO NO APP! NAO DA PARA CONTROLAR O FIRE STICK F
1482	BAIXEI E ATIVEI O AMAZON ALEXA, MAS NAO CONSIGO ATIVAR O COMANDO DE VOZ PARA FAI
1483	E BOM, MAS AINDA NAO ACHEI OPCAO DE ADAPTAR MEU ECHO PARA O PORTUGUES.

E vendo eles podemos ter uma ideia que talvez o celular de alguns clientes não seja tão otimizado, ou então a interface do usuário não tenha agradado tanto.

Dataset Negativo

view_negativo = ProfileReport(negativo)

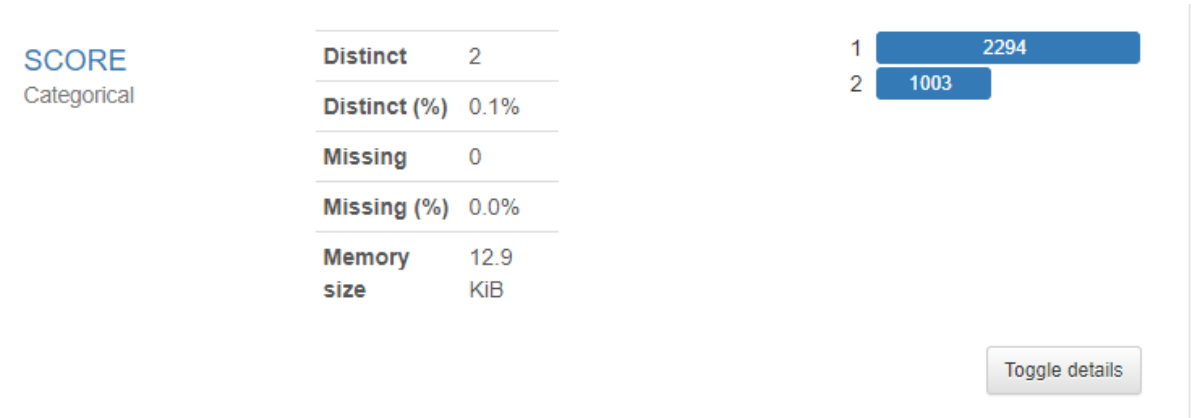
Com os dados negativos, temos as seguintes análises feitas:



Assim como foi nos comentários positivos, nos negativos ele consegue reconhecer o top 5 de palavras mais utilizadas entre os usuários, e sem grandes surpresas as cinco primeiras

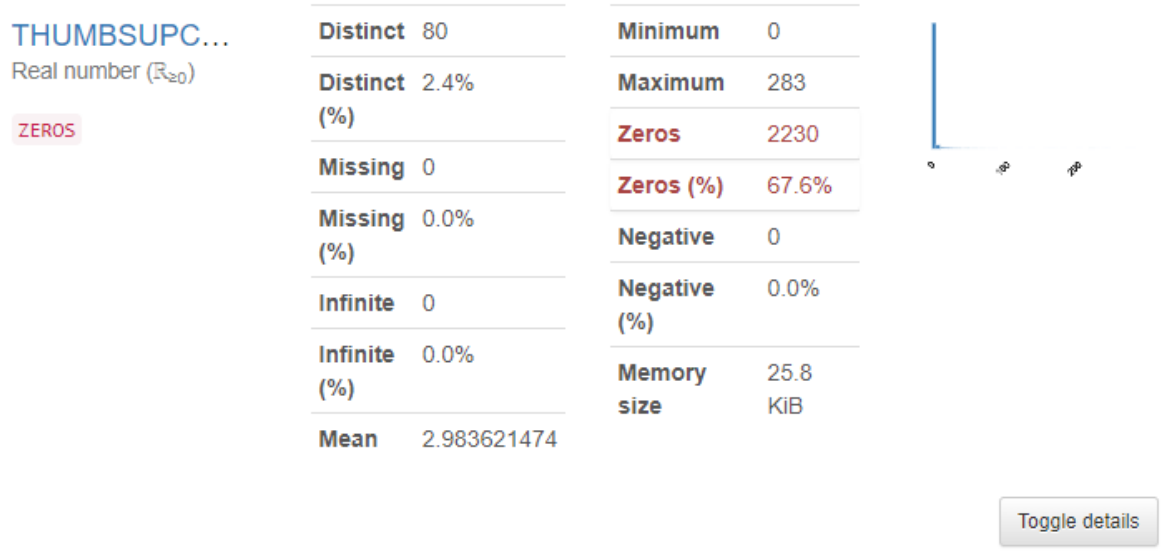
são “Horrível”, “Não funciona”, “Muito lento” e “Muito Ruim”, estranhamente há também a palavra “Bom”, o que nos mostra ser um sentimento positivo, o que pode demonstrar que talvez o usuário tenha inserido a pontuação erroneamente no momento da avaliação do aplicativo.

Analisando a coluna SCORE:



Diferentemente das avaliações positivas, em que havia um valor significantemente maior que o outro, nos negativos temos as variáveis mais balanceadas, com a tendência dos usuários votarem na menor pontuação possível do que a menos pior.

Na coluna de THUMBSUP também temos uma mudança visível:



Nos positivos temos 413 votos máximo para um único comentário, enquanto nos negativos apenas 283 foi o máximo, o que mostra mais uma relevância dos usuários com comentários positivos do que com os negativos.

E finalmente, ao analisar os 5 primeiros e últimos comentários temos uma situação um pouco anormal:

0	BOM
1	E BOM
2	O APLICATIVO TRAVA MUITO! NAO E MUITO FUNCIONAL.
3	NAO TOCA O ARTISTA QUE PEDIMOS, TOCA SEMPRE UMA RADIO ALEATORIA QUE NAO E DO ARTI
4	A M E I !!!!!!!!

3292	BOM APP MAS FALTA EM PORTUGUES
3293	NAO FUNCIONA TUDO, ROTINA E SKILLS POR EXEMPLO NAO ABRE SIMPLESMENTE TRAVA, TI
3294	I CANT DOWNLOAD SKILLS OR SET ROUTINES IN IT, THE APP DONT WORK
3295	APLICATIVO ESTA PESADO E TRAVANDO NO GALAXY 8 NOTES. UMA PENA.
3296	UMA MERD%

Há uma alta diferença de sentimento entre os comentários, sejam eles positivos e negativos visto, o que nos faz pensar que possivelmente alguns usuários podem ter avaliado errado o aplicativo com uma nota menor do que a desejada.

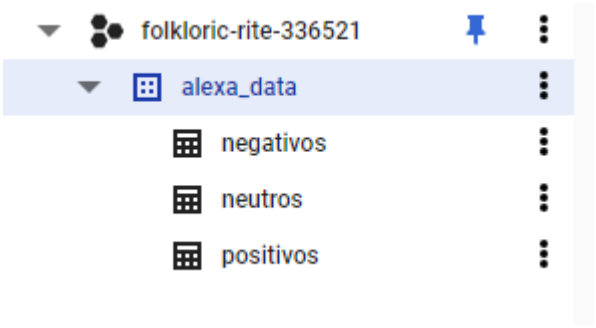
Conclusão

No total obtivemos 22578 dados analisados, o que torna mais da metade (79%) dos três datasets contendo valores positivos, enquanto os dados neutros (1483) e negativos (3296) formam menos da metade desse dataset, então nos mostra que a Alexa no Brasil está sendo bem recebida e avaliada pelos usuários.
E alguns desses dados negativos ainda acabaram sendo positivos, talvez pela falta de conhecimento no sistema de avaliação da Play Store, ou de interface de usuário.

Tarefa 02

Foi utilizado o BigQuery para realizar a tarefa.

Criei um banco chamado alexa_data, e três tabelas chamadas positivos, negativos e neutros, representando os dados obtidos anteriormente da requisição:



Cada tabela contém cinco colunas, permitindo nulos e dos tipos que variam entre integer, string e timestamp conforme é mostrado abaixo:

Nome do campo	Tipo	Modo	Tags de política	Descrição
CONTENT	STRING	NULLABLE		
SCORE	INTEGER	NULLABLE		
THUMBSUPCOUNT	INTEGER	NULLABLE		
REVIEWCREATEDVERSION	STRING	NULLABLE		
AT	TIMESTAMP	NULLABLE		

Cada arquivo foi inserido separadamente no banco, se atentando sempre com a extensão do arquivo e como ficou o formato dos dados em cada inserção.

Através do objeto json, obtemos a autenticação do BigQuery para poder fazer a inserção no banco e cria-lo.

key_path = "GBQ.json"

É criado também uma credencial com os serviços da plataforma do Google Cloud.

```
credentials = service_account.Credentials.from_service_account_file(
    key_path, scopes=["https://www.googleapis.com/auth/cloud-plataform"]
)
```

E em seguida e por último é feita a inserção dos dados de cada dataset para que seja feita a inserção dos mesmos no banco na Cloud.

```
for dataset, nome_tabela in zip(lista_dataset, lista_nome_tabela):
    dataset.to_gbq(project_id="testebigquery-336702",
                   destination_table="banco." + nome_tabela,
                   if_exists="replace")
```

Tarefa 03

Pipeline:

- [ ] Requisição dos dados fornecidos pelo aplicativo informado
- [ ] Preparação dos dados:
    - [ ] Feature Engineering
- [ ] Conversão dos dados para CSV
- [ ] Conexão com o banco
    - [ ] Verificação de tabelas existentes
    - [ ] Criação de nova tabela caso não exista
- [ ] Atualização do banco com os novos dados tratados