Jr Data Scientist - Evaluation -1

Part-1

1. Need to extract the numbers with an orange color background from the given text.
So I need to extract the values from the keys ‘id’ and ‘code’ from the dictionary which is given as a string.

Code : result=[int(i) for i in re.findall(r"['id''code']:(\d+)",st)] #st is the given text.
Output : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 648, 649, 650, 651, 652, 653, 3]

So the numbers in the result list which need to be extracted.

2. Goal : **To identify the reviews where the semantics of review text does not match rating**.

Given dataset consists of 7204 Samples and 10 features

Features:
{ID Review URL, Text, Star, Thumbs Up, User Name , Developer Reply, Version, Review Date, App ID}

Need to find a good text but the rating of that particular text doesn’t match.

Method:
As we need to find the text, here i have done preprocessing
For example: some of the text contains the emojis so I removed them and proceeded for further analysis on the raw text.

To find the required text, here I have taken all the samples which Star is less than and equal to 2 (because Rating doesn’t match the text)

Now I have used the pretrained sentiment analysis which is a transformers based model that is

```
from transformers import pipeline
sentiment_analysis=pipeline("sentiment-analysis")
sentiment_anlysis(input) = output
Input : Text
Output : list of score and label(POSITIVE or NEGATIVE)
```

Now I have applied above sentiment_analysis function on the text where star is <= 2

After applying I have considered the high score positive sentiment and am able to find the positive text with low rating.

From the results, most of the words used in the text are ‘Good’ for 39 times, ‘Nice’ for 15 times, 'Nice app’ for 6 times, and ‘Its a fantastic app’ etc..
Proceeding further I have deployed this using Flask and for the live link I have used SocketXP
And the livelink is https://ht17ms174-v5v8pzz6o2cnu8ae.socketxp.com.

3.Given Dataset consists of 3067 samples and 10 features
14 NULL values present in the Rank column of the data.

Features:
{**ID, Keyword , Rank, Country , Language, Date, App ID, Date of Last DescriptionChange, Short Description, Long Description**}

Used and required features Rank,Date,App ID, Date of Last Description Change.

Different keywords used for searching and their corresponding frequencies:

browser	608
android browser	608
privacy browser	370
privacy browsers for android	370
best privacy browsers	370
fast browser	370
ad free browser	370

Different App IDs appear after searching based on keywords and corresponding frequencies:

com.duckduckgo.mobile.android	740
com.cloudmosa.puffinTV	608
com.brave.browser	370
com.vivaldi.browser	370
com.transsion.phoenix	370
com.opera.browser	304
net.fast.web.browser	152
com.android.chrome	152

Ranks started from 1 to 119 and 134 that is no values for rank as mentioned null values present in data.

There is a correlation between the keyword rankings and Short Description
The App id 'com.vivaldi.browser' which has a short description:
Fast & private web browser with ad blocker, sync, dark & private mode

Its average Rank was 64 and keywords used for these app are
privacy browser , privacy browsers for android , best privacy browsers ,
fast browser , ad free browser

Needs to change the short description,because it may be due to the common words between the keyword used for searching and words in the short description.

Here App ID plays an important role in ranking because AppID is the unique representation of different apps so we know which app is performing well according to the rankings.

If you use keywords , 'privacy browsers for android','best privacy browsers', 'fast browser', 'ad free browser' for searching.
Apps like transsion.phoenix, vivaldi.browser are not getting as rank-1 these may get rank-1 if there is a chance of changing the Description.

As Chrome is the common browser mostly users use some analysis has done on it

- Google chrome as rank 1 for 18 times using keyword browser
- Google Chrome ranks in top 10 as an app when used keywords 'browser', 'android browser' 68 times.
- Its ranks 7 when android browser is used as keyword for 2 times
- The lowest rank was 20 and it got when browser and android browser used as keywords(mostly when android browser used as keyword.)

Here Date of Description change is also an important feature because
If say d is the date of Description change if we compare before d and after d some apps ranks were improved and also vice versa.

Part-2

Goal: If the sentence is grammatically correct or not.

Here I have used the pretrained model from hugging face which is trained with happy transformer on the dataset A Fluency Corpus and Benchmark for Grammatical Error Correction

```
from happytransformer import HappyTextToText
htt = HappyTextToText("T5", "vennify/t5-base-grammar-correction")
from happytransformer import TTSettings
args = TTSettings(num_beams=5, min_length=1)
htt.generate_text(input , args=args).text
```

Output : grammatically corrected version of text.

Text => Function() => Text

Here we need a label as output whether it is grammatically correct(1) or not (0)

So I have this applied this function htt.generate_text(input , args=args).text on the text and i compared this output text to input text if it matches i have given as label-1 otherwise label-0
Note: Here i have used label -2 because the function which is used giving multiple output of same kind when single words are given as input

So by assuming sentence is constructed with more than 3 words, i have used a condition if the text has less than 3 words the output is label-2 if it is more than 3 words the output is either 1 or 0.