wrangle_report

July 31, 2022

Wrangle Report

1. I've started from merging data together, since not all the tweets were in all datasets,I've rejected those they were not.

[1]:
```
# doggies2 = dogs2.merge(dogs3,on='tweet_id',how='inner')

# doggies = dogs.merge(doggies2,on='tweet_id',how='inner')
```

2. I wanted to work only with original tweets, not ones which were replayed to other users, so I've deleted those.

[2]:
```
# labels =doggies['in_reply_to_user_id'].dropna().index.to_list(),
# labels2 =doggies['retweeted_status_id'].dropna().index.to_list()
# doggies.drop(axis=0,labels=labels,inplace=True)
# doggies.drop(axis=0,labels=labels2,inplace=True)
```

3. Then I've noticed that some columns were empty, so I've deleted these columns. (the ones linked to retweets)

[3]:
```
#columns = ['in_reply_to_user_id','in_reply_to_status_id','retweeted_status_id',
#           'retweeted_status_user_id','retweeted_status_timestamp']

#doggies.drop(axis=1,columns=columns,inplace=True)
```

4. I've added new columns 'breed' and 'breed conf.' and used the script I've written to fill them with the values with the highest probability. NOTICE : In this code I'm also changing all breeds names to lowercase letters.

[4]:
```
#masterdog = doggies
#masterdog['breed'] = np.nan
#masterdog['breed conf.'] = np.nan
#indexer = []
#for _ in masterdog['tweet_id']:
#    indexer.append(_)
#masterdog.set_index('tweet_id',inplace=True)
#for _ in indexer:
#    p1 = masterdog.loc[_,'p1']
#    p2 = masterdog.loc[_,'p2']
#    p3 = masterdog.loc[_,'p3']
```

```
#    p1 = p1.lower()
#    p2 = p2.lower()
#    p3 = p3.lower()

#    p1_conf = masterdog.loc[_,'p1_conf']
#    p2_conf = masterdog.loc[_,'p2_conf']
#    p3_conf = masterdog.loc[_,'p3_conf']

#    p1_dog = masterdog.loc[_,'p1_dog']
#    p2_dog = masterdog.loc[_,'p2_dog']
#    p3_dog = masterdog.loc[_,'p3_dog']

#    if p1_dog == True:
#        masterdog.loc[_,'breed'] = p1
#        masterdog.loc[_,'breed conf.'] = p1_conf
#    elif p2_dog == True and p1_dog != True:
#        masterdog.loc[_,'breed'] = p2
#        masterdog.loc[_,'breed conf.'] = p2_conf
#    elif p3_dog == True and p1_dog != True and p2_dog != True:
#        masterdog.loc[_,'breed'] = p3
#        masterdog.loc[_,'breed conf.'] = p3_conf
#    else:
#        continue
```

5. Since I had got different denominators, I've gotten rid of entries not equal to 10, so Our rating will be more accurate.

[5]:
```
#masterdog = masterdog.where(masterdog['rating_denominator']==10).dropna()
```

6. I've also gotten rid of data We won't be using further. Like columns used before in script to predict breed of dog. And some others which had many values marked as None.

[6]:
```
#masterdog.drop(axis=1,columns=['p1','p1_conf','p1_dog','p2','p2_conf','p2_dog',
#                               'p3','p3_conf','p3_dog'],inplace=True)
```

7. Some names in dataset are missing. We don't want to drop records just becouse names were extracted wrongly, but from We can see in description some of tweets are still not dogs. We will delete them.

[1]:
```
#masterdog.reset_index(inplace=True)
#labels3 = []
#for _ in masterdog['tweet_id']:
#    labels3.append(_)
#masterdog.set_index('tweet_id',inplace=True)
#toTrash = []
#for _ in labels3:
#    tabu = 'we only rate dogs'
```

```
#    text =  masterdog.loc[_,'text']
#    text = text.lower()
#    if tabu in text:
#        toTrash.append(_)
#    else:
#        continue
```

8. We will delete columns doggo, floofer, pupper, puppo. Becouse I'm not interested in them in my analyses.

[ ]:
```
#toTrash2 = ['doggo', 'floofer', 'pupper', 'puppo']
#masterdog.drop(axis=1,columns=toTrash2,inplace=True)
```

9. Then I had to discard breeds which occurs less than 20 times and marked them as others just for sanity and visibility of plots.

[7]:
```
#data = masterdog['breed'].value_counts()

#breeds = pd.DataFrame(data)
#breeds.reset_index(inplace=True)
#indexer2 = []
#for _ in breeds['index']:
#    indexer2.append(_)
#breeds.set_index('index',inplace=True)

#other = 0
#for _ in indexer2:
#    race = breeds.loc[_,'breed']
#    if race less then 20:
#        other += 1
#        breeds = breeds.drop(axis=0,labels=_).dropna()
#    else:
#        continue
#breeds.reset_index(inplace=True)
#newline = {'index':'other','breed':94}
#plotbreeds = breeds.append(newline,ignore_index=True)
```

[ ]:
```
```