PN_balance

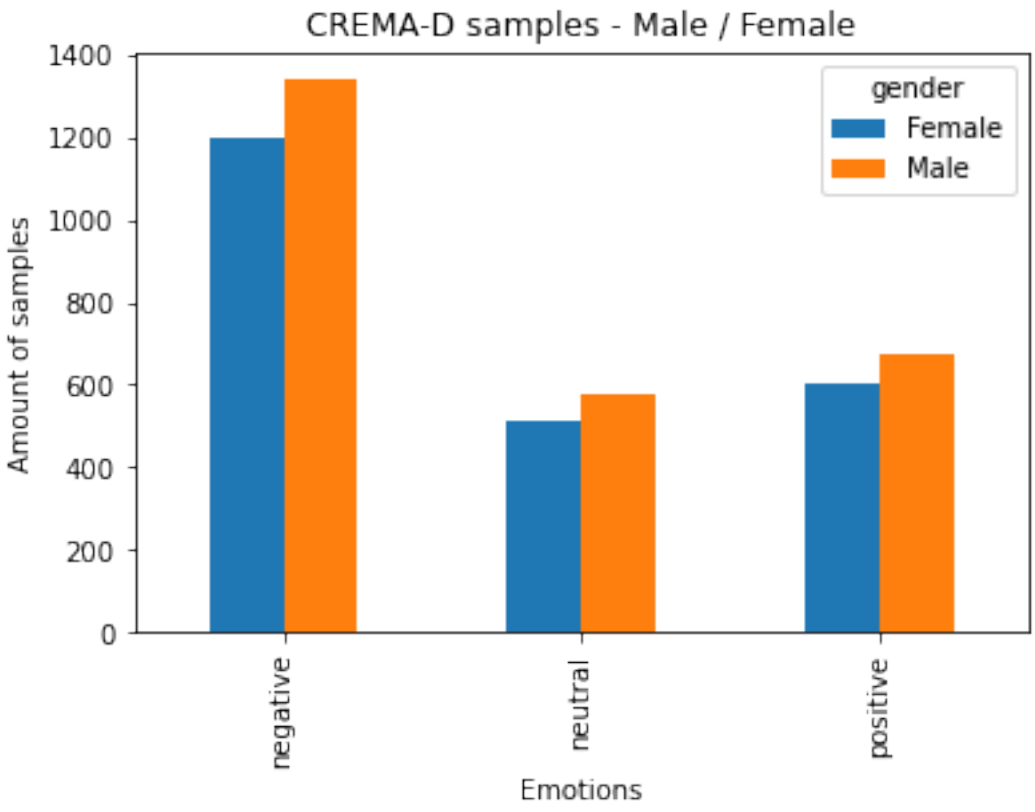
January 10, 2022

[1]:
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time
from imblearn.combine import SMOTETomek
from imblearn.under_sampling import NearMiss
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
```

[2]:
```
%run /data/emo/notebooks/source/pipeline/dataset_loader.ipynb
```

1 CremaBinair

[3]:
```
# Histo Pos - Neg - Neu
df_male = MaleSplitCremaBinair.load_dataset()
df_female = FemaleSplitCremaBinair.load_dataset()

df = pd.concat([df_male, df_female])

df.groupby(['emotion', 'gender']).size().unstack(level=1).plot(kind='bar')
plt.title('CREMA-D samples - Male / Female')
plt.ylabel('Amount of samples')
plt.xlabel('Emotions')
plt.show()

print(df_male.groupby('emotion').size())
print("")
print(df_female.groupby('emotion').size())
```




```
emotion
negative    1341
neutral      575
positive     671
dtype: int64

emotion
negative    1199
neutral      512
positive     600
dtype: int64
```

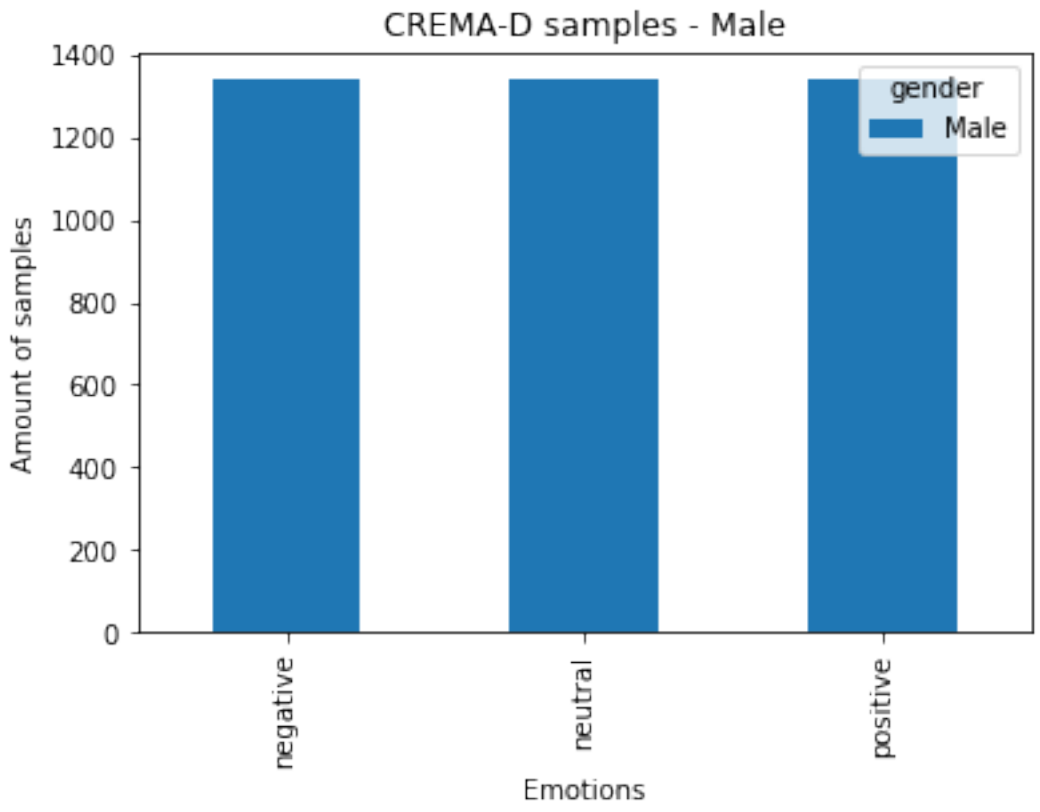
[4]:
```
# Oversampeling RandomSampler Male

# https://github.com/ufoym/imbalanced-dataset-sampler

# https://www.youtube.com/watch?v=0JedgzdipC0&ab_channel=KrishNaik
# https://github.com/krishnaik06/Handle-Imbalanced-Dataset/blob/master/
 ↪Handling%20Imbalanced%20Data-%20Over%20Sampling.ipynb

# https://imbalanced-learn.org/stable/over_sampling.html
```

```
columns = df_male.columns.tolist()
columns = [c for c in columns if c not in ["emotion"]]

X = df_male[columns]
Y = df_male['emotion']

# Oversampeling
male_grouped_pos = df_male[df_male['emotion'] == 'positive']
male_grouped_neu = df_male[df_male['emotion'] == 'neutral']
male_grouped_neg = df_male[df_male['emotion'] == 'negative']

print(f"Pos:{male_grouped_pos.shape}, Neu:{male_grouped_neu.shape}, Neg:␣
 ↪{male_grouped_neg.shape}")
print(f"Original X:{X.shape}, Y:{Y.shape}")

ros = RandomOverSampler()
X_train_res, y_train_res = ros.fit_resample(X, Y)

y_train_pos = y_train_res[y_train_res == 'positive']
y_train_neu = y_train_res[y_train_res == 'neutral']
y_train_neg = y_train_res[y_train_res == 'negative']

print(f"Pos:{y_train_pos.shape}, Neu:{y_train_neu.shape}, Neg: {y_train_neg.
 ↪shape}")
print(f"Resampled X:{X_train_res.shape}, Y:{y_train_res.shape}")


print(X_train_res)
print(y_train_res)

X_train_res['emotion'] = y_train_res
```

```
Pos:(671, 4), Neu:(575, 4), Neg: (1341, 4)
Original X:(2587, 3), Y:(2587,)
Pos:(1341,), Neu:(1341,), Neg: (1341,)
Resampled X:(4023, 3), Y:(4023,)
     gender subset                                              file_path
0      Male   None  /data/emo/notebooks/source/datasets/crema/1023…
1      Male   None  /data/emo/notebooks/source/datasets/crema/1001…
2      Male   None  /data/emo/notebooks/source/datasets/crema/1040…
3      Male   None  /data/emo/notebooks/source/datasets/crema/1034…
4      Male   None  /data/emo/notebooks/source/datasets/crema/1035…
…       …     …                                                  …
4018   Male   None  /data/emo/notebooks/source/datasets/crema/1086…
4019   Male   None  /data/emo/notebooks/source/datasets/crema/1038…
4020   Male   None  /data/emo/notebooks/source/datasets/crema/1051…
```

```
4021   Male   None  /data/emo/notebooks/source/datasets/crema/1031…
4022   Male   None  /data/emo/notebooks/source/datasets/crema/1069…

[4023 rows x 3 columns]
0       negative
1        neutral
2       negative
3        neutral
4       negative
          …
4018    positive
4019    positive
4020    positive
4021    positive
4022    positive
Name: emotion, Length: 4023, dtype: object
```

[5]:
```
# Histo Male
df = X_train_res

df.groupby(['emotion', 'gender']).size().unstack(level=1).plot(kind='bar')
plt.title('CREMA-D samples - Male')
plt.ylabel('Amount of samples')
plt.xlabel('Emotions')
plt.show()

print(df.groupby('emotion').size())
```




```
emotion
negative    1341
neutral     1341
positive    1341
dtype: int64
```

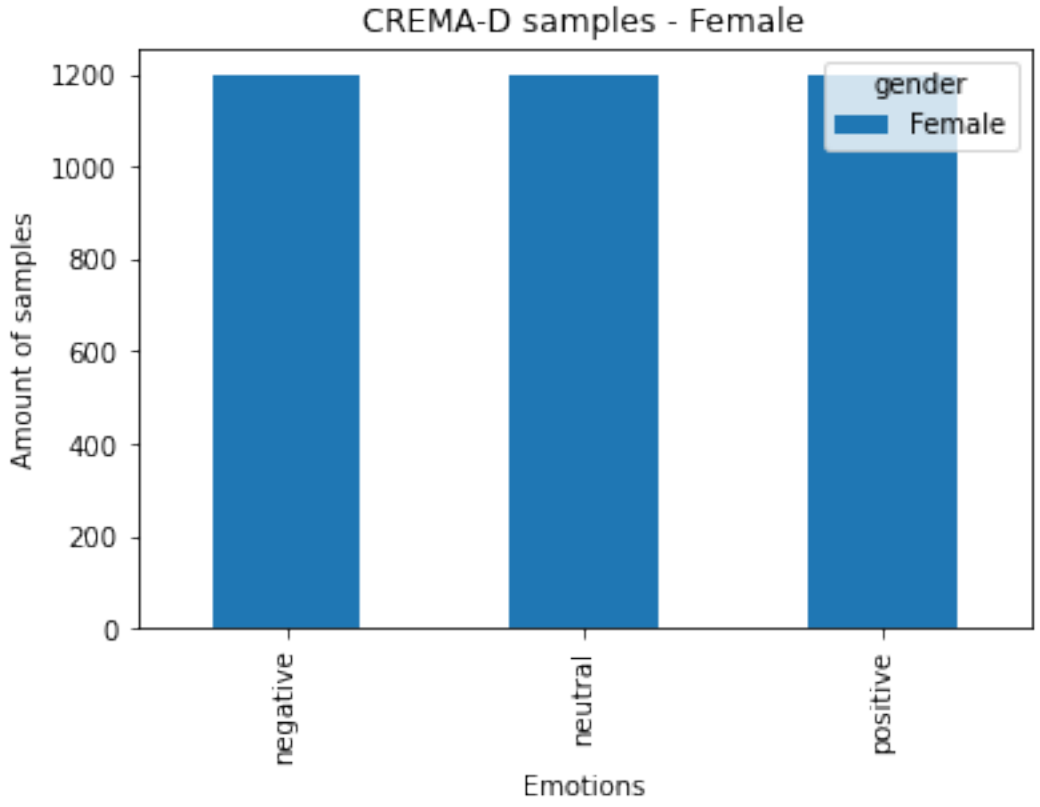
[6]:
```python
# Oversampeling RandomSampler Female
columns = df_female.columns.tolist()
columns = [c for c in columns if c not in ["emotion"]]

X = df_female[columns]
Y = df_female['emotion']

# Oversampeling
female_grouped_pos = df_female[df_female['emotion'] == 'positive']
female_grouped_neu = df_female[df_female['emotion'] == 'neutral']
female_grouped_neg = df_female[df_female['emotion'] == 'negative']

print(f"Pos:{female_grouped_pos.shape}, Neu:{female_grouped_neu.shape}, Neg:␣
 ↪{female_grouped_neg.shape}")
print(f"Original X:{X.shape}, Y:{Y.shape}")
```

```
ros = RandomOverSampler()
X_train_res, y_train_res = ros.fit_resample(X, Y)

y_train_pos = y_train_res[y_train_res == 'positive']
y_train_neu = y_train_res[y_train_res == 'neutral']
y_train_neg = y_train_res[y_train_res == 'negative']

print(f"Pos:{y_train_pos.shape}, Neu:{y_train_neu.shape}, Neg: {y_train_neg.
 ↪shape}")
print(f"Resampled X:{X_train_res.shape}, Y:{y_train_res.shape}")


print(X_train_res)
print(y_train_res)

X_train_res['emotion'] = y_train_res
```

```
Pos:(600, 4), Neu:(512, 4), Neg: (1199, 4)
Original X:(2311, 3), Y:(2311,)
Pos:(1199,), Neu:(1199,), Neg: (1199,)
Resampled X:(3597, 3), Y:(3597,)
      gender subset                                          file_path
0     Female   None  /data/emo/notebooks/source/datasets/crema/1037…
1     Female   None  /data/emo/notebooks/source/datasets/crema/1058…
2     Female   None  /data/emo/notebooks/source/datasets/crema/1054…
3     Female   None  /data/emo/notebooks/source/datasets/crema/1003…
4     Female   None  /data/emo/notebooks/source/datasets/crema/1007…
…      …    …                                                    …
3592  Female   None  /data/emo/notebooks/source/datasets/crema/1082…
3593  Female   None  /data/emo/notebooks/source/datasets/crema/1013…
3594  Female   None  /data/emo/notebooks/source/datasets/crema/1061…
3595  Female   None  /data/emo/notebooks/source/datasets/crema/1013…
3596  Female   None  /data/emo/notebooks/source/datasets/crema/1009…

[3597 rows x 3 columns]
0       negative
1       negative
2        neutral
3       negative
4       negative
          …
3592    positive
3593    positive
3594    positive
3595    positive
3596    positive
```

```
Name: emotion, Length: 3597, dtype: object
```

[7]:
```
# Histo Female
df = X_train_res

df.groupby(['emotion', 'gender']).size().unstack(level=1).plot(kind='bar')
plt.title('CREMA-D samples - Female')
plt.ylabel('Amount of samples')
plt.xlabel('Emotions')
plt.show()

print(df.groupby('emotion').size())
```




```
emotion
negative    1199
neutral     1199
positive    1199
dtype: int64
```

[8]:
```
# Oversampeling SMOTETomek Male
```

```
# columns = df_male.columns.tolist()
# columns = [c for c in columns if c not in ["emotion"]]

# X = df_male[columns]
# Y = df_male['emotion']
# print(X)

# male_grouped_pos = df_male[df_male['emotion'] == 'positive']
# male_grouped_neu = df_male[df_male['emotion'] == 'neutral']
# male_grouped_neg = df_male[df_male['emotion'] == 'negative']

# print(f"Pos:{male_grouped_pos.shape}, Neu:{male_grouped_neu.shape}, Neg:␣
 ↪{male_grouped_neg.shape}")
# print(f"Original X:{X.shape}, Y:{Y.shape}")

# smk = SMOTETomek()
# X_res,y_res=smk.fit_resample(X,Y)

# y_train_pos = y_train_res[y_train_res == 'positive']
# y_train_neu = y_train_res[y_train_res == 'neutral']
# y_train_neg = y_train_res[y_train_res == 'negative']

# X_res.shape,y_res.shape

# print(f"Pos:{y_train_pos.shape}, Neu:{y_train_neu.shape}, Neg: {y_train_neg.
 ↪shape}")
# print(f"Resampled X:{X_train_res.shape}, Y:{y_train_res.shape}")

# print(X_train_res)
# print(y_train_res)

# X_train_res['emotion'] = y_train_res
```

2 QuaternairCombinedPN

[10]:
```
df = QuaternairCombinedPN.load_dataset()

df.groupby(['emotion', 'gender']).size().unstack(level=1).plot(kind='bar')
plt.title('QuaternairCombinedPN samples')
plt.ylabel('Amount of samples')
plt.xlabel('Emotions')
plt.show()

print(df.groupby(['emotion', 'gender']).size())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_11595/2278170828.py in <module>
----> 1 df = QuaternairCombinedPN.load_dataset()
      2
      3 df.groupby(['emotion', 'gender']).size().unstack(level=1).
 ↪plot(kind='bar')
      4 plt.title('QuaternairCombinedPN samples')
      5 plt.ylabel('Amount of samples')

/tmp/ipykernel_11595/1532011642.py in load_dataset(cls)
     20                 value['emotion'] = 'neutral'
     21
---> 22             components = np.array([value, "Female", None, os.path.
 ↪join(path, file)])
     23             components.append(component)
     24

NameError: name 'path' is not defined
```

[ ]:
```
# Oversampeling RandomSampler

columns = df.columns.tolist()
columns = [c for c in columns if c not in ["emotion"]]

X = df[columns]
Y = df['emotion']

#Oversampeling
gender_grouped_pos = df[df['emotion'] == 'positive']
gender_grouped_neu = df[df['emotion'] == 'neutral']
gender_grouped_neg = df[df['emotion'] == 'negative']

print(f"Pos:{gender_grouped_pos.shape}, Neu:{gender_grouped_neu.shape}, Neg:␣
 ↪{gender_grouped_neg.shape}")
print(f"Original X:{X.shape}, Y:{Y.shape}")

ros = RandomOverSampler()
X_train_res, y_train_res = ros.fit_resample(X, Y)

y_train_pos = y_train_res[y_train_res == 'positive']
y_train_neu = y_train_res[y_train_res == 'neutral']
y_train_neg = y_train_res[y_train_res == 'negative']

print(f"Pos:{y_train_pos.shape}, Neu:{y_train_neu.shape}, Neg: {y_train_neg.
 ↪shape}")
```

```
print(f"Resampled X:{X_train_res.shape}, Y:{y_train_res.shape}")


print(X_train_res)
print(y_train_res)

X_train_res['emotion'] = y_train_res
```

[ ]:
```
df = X_train_res

df.groupby(['emotion', 'gender']).size().unstack(level=1).plot(kind='bar')
plt.title('QuaternairCombinedPN samples')
plt.ylabel('Amount of samples')
plt.xlabel('Emotions')
plt.show()

print(df.groupby(['emotion', 'gender']).size())
```

[ ]:
```

```