dataset_loader

January 11, 2022

available_columns = 'emotion', 'gender', 'subset', 'file_path'

[1]:
```
import os
import numpy as np
import pandas as pd

class Loader:
    identifier = None

    @classmethod # returns dataframe with dataset info
    def load_dataset(cls): return None
```

1 Load CREMA-D Normal

[2]:
```
class NormalCrema(Loader):
    identifier = 'crema_normal'

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/crema'
        for file in os.listdir(path):
            component = np.array(file.replace('.', '_').split('_'))
            component = np.array([component[2], None, None, os.path.join(path,␣
↪file)])
            components.append(component)
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

2 Load CREMA-D Splitted

[3]:
```
class SplittedCrema(Loader):
    identifier = 'crema_splitted'

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/crema_splitted'
        subsets = ['Test', 'Train', 'Validate']
        for subset in subsets:
            subset_path = f'{path}/{subset}'
            for file in os.listdir(subset_path):
                component = np.array(file.replace('.', '_').split('_'))
                component = np.array([component[2], None, subset, os.path.
↪join(subset_path, file)])
                components.append(component)
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

3 Load RAVDESS Normal

[4]:
```
class NormalRavdess(Loader):
    identifier = 'ravdess_normal'
    _emotion_labels = {
      '01':'neutral',
      '02':'calm',
      '03':'happy',
      '04':'sad',
      '05':'angry',
      '06':'fearful',
      '07':'disgust',
      '08':'surprised'
    }

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/ravdess'
        for file in os.listdir(path):
            component = np.array(file.replace('.', '-').split('-'))
            component = np.array([cls._emotion_labels[component[2]], None,␣
↪None, os.path.join(path, file)])
            components.append(component)
```

```
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

4 Load RAVDESS Splitted

[5]:
```
class SplittedRavdess(Loader):
    identifier = 'ravdess_splitted'
    _emotion_labels = {
      '01':'neutral',
      '02':'calm',
      '03':'happy',
      '04':'sad',
      '05':'angry',
      '06':'fearful',
      '07':'disgust',
      '08':'surprised'
    }

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/ravdess_splitted'
        subsets = ['Test', 'Train', 'Validate']
        for subset in subsets:
            subset_path = f'{path}/{subset}'
            for file in os.listdir(subset_path):
                component = np.array(file.replace('.', '-').split('-'))
                component = np.array([cls._emotion_labels[component[2]], None,␣
↪subset, os.path.join(subset_path, file)])
                components.append(component)
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

5 Load CREMA-D Binair

[6]:
```
class BinairCrema(Loader):
    identifier = 'crema_binair'

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/crema'
        for file in os.listdir(path):
```

```
            component = np.array(file.replace('.', '_').split('_'))
            polarity = 'negative' if component[2] in ['ANG', 'SAD', 'FEA',␣
↪'DIS'] else 'positive'
            component = np.array([polarity, None, None, os.path.join(path,␣
↪file)])
            components.append(component)
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

6 Load Male CREMA-D Binair

[7]:
```
class MaleSplitCremaBinair(Loader):
    identifier = 'crema_male'
    _crema_d_female_samples =␣
↪[1002,1003,1004,1006,1007,1008,1009,1010,1012,1013,1018,1020,1021,1024,1025,1028,1029,1030,
        ␣
↪1052,1053,1054,1055,1056,1058,1060,1061,1063,1072,1073,1074,1075,1076,1078,1079,1082,1084,1

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/crema'
        for file in os.listdir(path):
            component = np.array(file.replace('.', '_').split('_'))

            if int(component[0]) in cls._crema_d_female_samples:
                continue

            if component[2] not in ['ANG', 'SAD', 'HAP', 'NEU']:
                continue

            polarity = ""
            if component[2] in ['ANG', 'SAD']:
                polarity = 'negative'
            elif component[2] in ['HAP']:
                polarity = 'positive'
            elif component[2] in ['NEU']:
                polarity = 'neutral'

            component = np.array([polarity, "Male", None, os.path.join(path,␣
↪file)])
            components.append(component)

        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

7 Load Female CREMA-D Binair

[8]:
```
class FemaleSplitCremaBinair(Loader):
    identifier = 'crema_male'
    _crema_d_female_samples =␣
↪[1002,1003,1004,1006,1007,1008,1009,1010,1012,1013,1018,1020,1021,1024,1025,1028,1029,1030,
        ␣
↪1052,1053,1054,1055,1056,1058,1060,1061,1063,1072,1073,1074,1075,1076,1078,1079,1082,1084,1

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/crema'
        for file in os.listdir(path):
            component = np.array(file.replace('.', '_').split('_'))

            if int(component[0]) not in cls._crema_d_female_samples:
                continue

            if component[2] not in ['ANG', 'SAD', 'HAP', 'NEU']:
                continue

            polarity = ""
            if component[2] in ['ANG', 'SAD']:
                polarity = 'negative'
            elif component[2] in ['HAP']:
                polarity = 'positive'
            elif component[2] in ['NEU']:
                polarity = 'neutral'

            component = np.array([polarity, "Female", None, os.path.join(path,␣
↪file)])
            components.append(component)

        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

8 Load RAVDESS Binair

TODO: Split Male and Female

[9]:
```
class BinairRavdess(Loader):
    identifier = 'ravdess_binair'

    @classmethod
```

```
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/ravdess'
        for file in os.listdir(path):
            component = np.array(file.replace('.', '_').split('_'))
            polarity = 'negative' if component[2] in ['ANG', 'SAD', 'FEA',␣
↪'DIS'] else 'positive'
            component = np.array([polarity, None, None, os.path.join(path,␣
↪file)])
            components.append(component)
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

9 Load Male RAVDESS Binair

[10]:
```
class MaleBinairRavdess(Loader):
    identifier = 'ravdess_postive_negative_male'
    _emotion_labels = {
      '01':'"neutral"',
      '02':'calm',
      '03':'happy',
      '04':'sad',
      '05':'angry',
      '06':'fearful',
      '07':'disgust',
      '08':'surprised'
    }

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/ravdess'

        for file in os.listdir(path):
            component = np.array(file.replace('.', '-').split('-'))

            if int(component[6]) % 2 == 0:
                continue

            if component[2] not in ['01', '02', '03', '04', '05']:
                continue

            polarity = ""
            if component[2] in ['05', '04']:
                polarity = 'negative'
```

```
            elif component[2] in ['03', '02']:
                polarity = 'positive'
            elif component[2] in ['01']:
                polarity = 'neutral'

            component = np.array([polarity, "Male", None, os.path.join(path,␣
↪file)])
            components.append(component)

        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

10 Load Female RAVDESS Binair

[11]:
```
class FemaleBinairRavdess(Loader):
    identifier = 'ravdess_postive_negative_female'
    _emotion_labels = {
      '01':'"neutral"',
      '02':'calm',
      '03':'happy',
      '04':'sad',
      '05':'angry',
      '06':'fearful',
      '07':'disgust',
      '08':'surprised'
    }

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/ravdess'

        for file in os.listdir(path):
            component = np.array(file.replace('.', '-').split('-'))

            if int(component[6]) % 2 == 0:
                continue

            if component[2] not in ['01', '02', '03', '04', '05']:
                continue

            polarity = ""
            if component[2] in ['05', '04']:
                polarity = 'negative'
            elif component[2] in ['03', '02']:
```

```
            polarity = 'positive'
        elif component[2] in ['01']:
            polarity = 'neutral'

        component = np.array([polarity, "Female", None, os.path.join(path,␣
↪file)])
        components.append(component)

    return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

11 Load SAVEE Normal

[12]:
```
class NormalSavee(Loader):
    identifier = 'savee_normal'

    @classmethod
    def load_dataset(cls):
        emotion = []
        gender = []
        paths = []
        subset = []
        path = '/data/emo/notebooks/source/datasets/savee/'
        for i in os.listdir(path):
            if i[-8:-6] == '_a': emotion.append('angry')
            elif i[-8:-6] == '_d': emotion.append('disgust')
            elif i[-8:-6] == '_f': emotion.append('fear')
            elif i[-8:-6] == '_h': emotion.append('happy')
            elif i[-8:-6] == '_n': emotion.append('neutral')
            elif i[-8:-6] == 'sa': emotion.append('sad')
            elif i[-8:-6] == 'su': emotion.append('surprise')
            paths.append(path + i)
            gender.append('male')
            subset.append(None)
        SAVEE_df = pd.DataFrame(emotion, columns=['emotion'])
        SAVEE_df = pd.concat([SAVEE_df, pd.DataFrame(gender,␣
↪columns=['gender'])], axis=1)
        SAVEE_df = pd.concat([SAVEE_df, pd.DataFrame(subset,␣
↪columns=['subset'])], axis=1)
        SAVEE_df = pd.concat([SAVEE_df, pd.DataFrame(paths,␣
↪columns=['file_path'])], axis=1)

        return pd.DataFrame(data=SAVEE_df,␣
↪columns=['emotion','gender','subset','file_path'])
```

12 Load SAVEE Splitted

[13]:
```
class SplittedSavee(Loader):
    identifier = 'savee_splitted'

    @classmethod
    def load_dataset(cls):
        emotion = []
        gender = []
        paths = []
        subsetx = []
        path = '/data/emo/notebooks/source/datasets/savee_splitted'
        subsets = ['Test', 'Train', 'Validate']
        for subset in subsets:
            subset_path = f'{path}/{subset}'
            for i in os.listdir(subset_path):
                if i[-8:-6] == '_a':
                    emotion.append('angry')
                elif i[-8:-6] == '_d':
                    emotion.append('disgust')
                elif i[-8:-6] == '_f':
                    emotion.append('fear')
                elif i[-8:-6] == '_h':
                    emotion.append('happy')
                elif i[-8:-6] == '_n':
                    emotion.append('neutral')
                elif i[-8:-6] == 'sa':
                    emotion.append('sad')
                elif i[-8:-6] == 'su':
                    emotion.append('surprise')
                else:
                    emotion.append('error')
                paths.append(f'{subset_path}/{i}')
                gender.append('male')
                subsetx.append(subset)
        SAVEE_df = pd.DataFrame(emotion, columns=['emotion'])
        SAVEE_df = pd.concat([SAVEE_df, pd.DataFrame(gender,␣
↪columns=['gender'])], axis=1)
        SAVEE_df = pd.concat([SAVEE_df, pd.DataFrame(subsetx,␣
↪columns=['subset'])], axis=1)
        SAVEE_df = pd.concat([SAVEE_df, pd.DataFrame(paths,␣
↪columns=['file_path'])], axis=1)
        return pd.DataFrame(data=SAVEE_df,␣
↪columns=['emotion','gender','subset','file_path'])
```

13 Load TESS Normal

[14]:
```
class NormalTess(Loader):
    identifier = 'tess_normal'

    @classmethod
    def load_dataset(cls):
        emotion = []
        gender = []
        paths = []
        subset = []
        path = '/data/emo/notebooks/source/datasets/tess/'
        for i in os.listdir(path):
            fname = os.listdir(path + i)
            for f in fname:
                if i == 'OAF_angry' or i == 'YAF_angry':
                    emotion.append('angry')
                elif i == 'OAF_disgust' or i == 'YAF_disgust':
                    emotion.append('disgust')
                elif i == 'OAF_Fear' or i == 'YAF_fear':
                    emotion.append('fear')
                elif i == 'OAF_happy' or i == 'YAF_happy':
                    emotion.append('happy')
                elif i == 'OAF_neutral' or i == 'YAF_neutral':
                    emotion.append('neutral')
                elif i == 'OAF_Pleasant_surprise' or i ==␣
↪'YAF_pleasant_surprised':
                    emotion.append('surprise')
                elif i == 'OAF_Sad' or i == 'YAF_sad':
                    emotion.append('sad')
                else:
                    emotion.append('Unknown')
                paths.append(path + i + "/" + f)
                gender.append('female')
                subset.append(None)
        TESS_df = pd.DataFrame(emotion, columns=['emotion'])
        TESS_df = pd.concat([TESS_df, pd.DataFrame(gender,␣
↪columns=['gender'])], axis=1)
        TESS_df = pd.concat([TESS_df, pd.DataFrame(subset,␣
↪columns=['subset'])], axis=1)
        TESS_df = pd.concat([TESS_df, pd.DataFrame(paths,␣
↪columns=['file_path'])], axis=1)

        return pd.DataFrame(data=TESS_df,␣
↪columns=['emotion','gender','subset','file_path'])
```

14 Load TESS Splitted

[15]:
```
class SplittedTess(Loader):
    identifier = 'savee_splitted'

    @classmethod
    def load_dataset(cls):
        emotion = []
        gender = []
        paths = []
        subsetx = []
        path = '/data/emo/notebooks/source/datasets/tess_splitted'
        subsets = ['Test', 'Train', 'Validate']
        for subset in subsets:
            subset_path = f'{path}/{subset}'
            for i in os.listdir(subset_path):
                if 'angry' in i:
                    emotion.append('angry')
                elif 'disgust' in i:
                    emotion.append('disgust')
                elif 'fear' in i:
                    emotion.append('fear')
                elif 'happy' in i:
                    emotion.append('happy')
                elif 'neutral' in i:
                    emotion.append('neutral')
                elif 'suprised' in i:
                    emotion.append('surprise')
                elif 'sad' in i:
                    emotion.append('sad')
                else:
                    emotion.append('Unknown')
                paths.append(f'{subset_path}/{i}')
                gender.append('female')
                subsetx.append(subset)
        TESS_df = pd.DataFrame(emotion, columns=['emotion'])
        TESS_df = pd.concat([TESS_df, pd.DataFrame(gender,␣
↪columns=['gender'])], axis=1)
        TESS_df = pd.concat([TESS_df, pd.DataFrame(subsetx,␣
↪columns=['subset'])], axis=1)
        TESS_df = pd.concat([TESS_df, pd.DataFrame(paths,␣
↪columns=['file_path'])], axis=1)
        return pd.DataFrame(data=TESS_df,␣
↪columns=['emotion','gender','subset','file_path'])
```

15 Load Quaternair Combined

[16]:
```
class QuaternairCombined(Loader):
    identifier = 'combined_quaternair'

    @classmethod
    def load_dataset(cls):
        loaded_dataset_1 = NormalCrema.load_dataset()
        loaded_dataset_2 = NormalRavdess.load_dataset()
        loaded_dataset_3 = NormalSavee.load_dataset()
        loaded_dataset_4 = NormalTess.load_dataset()
        dataset = pd.concat([loaded_dataset_1, loaded_dataset_2,
                             loaded_dataset_3, loaded_dataset_4],␣
↪ignore_index=True, sort=False)
        for index, value in dataset.iterrows():
            if value['emotion'] in ['SAD', 'sad']: value['emotion'] = 'sad'
            if value['emotion'] in ['ANG', 'angry']: value['emotion'] = 'angry'
            if value['emotion'] in ['NEU', 'neutral']: value['emotion'] =␣
↪'neutral'
            if value['emotion'] in ['HAP', 'happy']: value['emotion'] = 'happy'
        return dataset.loc[dataset['emotion'].isin(['angry', 'neutral',␣
↪'happy', 'sad'])]
```

16 Load Trinair Combined

[17]:
```
class TrinairCombinedPN(Loader):
    identifier = 'combined_trinair'

    @classmethod
    def load_dataset(cls):
        loaded_dataset_1 = NormalCrema.load_dataset()
        loaded_dataset_2 = NormalRavdess.load_dataset()
        loaded_dataset_3 = NormalSavee.load_dataset()
        loaded_dataset_4 = NormalTess.load_dataset()
        dataset = pd.concat([loaded_dataset_1, loaded_dataset_2,
                             loaded_dataset_3, loaded_dataset_4],␣
↪ignore_index=True, sort=False)
        for index, value in dataset.iterrows():
            if value['emotion'] in ['ANG', 'angry', 'SAD', 'sad']:
                value['emotion'] = 'negative'
            elif value['emotion'] in ['HAP', 'happy', 'CAL', 'calm']:
                value['emotion'] = 'positive'
            elif value['emotion'] in ['NEU', 'neutral']:
                value['emotion'] = 'neutral'
```

```
        return dataset.loc[dataset['emotion'].isin(['negative', 'positive',␣
↪'neutral'])]
```

17 Load Quaternair Combined Splitted

[18]:
```
class QuaternairCombinedSplitted(Loader):
    identifier = 'combined_quaternair_splitted'

    @classmethod
    def load_dataset(cls):
        loaded_dataset_1 = SplittedCrema.load_dataset()
        loaded_dataset_2 = SplittedRavdess.load_dataset()
        loaded_dataset_3 = SplittedSavee.load_dataset()
        loaded_dataset_4 = SplittedTess.load_dataset()
        dataset = pd.concat([loaded_dataset_1, loaded_dataset_2,
                             loaded_dataset_3, loaded_dataset_4],␣
↪ignore_index=True, sort=False)
        for index, value in dataset.iterrows():
            if value['emotion'] in ['SAD', 'sad']: value['emotion'] = 'sad'
            if value['emotion'] in ['ANG', 'angry']: value['emotion'] = 'angry'
            if value['emotion'] in ['NEU', 'neutral']: value['emotion'] =␣
↪'neutral'
            if value['emotion'] in ['HAP', 'happy']: value['emotion'] = 'happy'
        return dataset.loc[dataset['emotion'].isin(['angry', 'neutral',␣
↪'happy', 'sad'])]
```

18 Load Male Quaternair Combined

[19]:
```
class QuaternairMaleCombinedPN(Loader):
    identifier = 'combined_quaternair_Male_Positive_Negative'

    @classmethod
    def load_dataset(cls):
        components = []
        loaded_dataset_1 = NormalCrema.load_dataset()
        loaded_dataset_2 = NormalRavdess.load_dataset()
        loaded_dataset_3 = NormalSavee.load_dataset()
        loaded_dataset_4 = NormalTess.load_dataset()
        dataset = pd.concat([loaded_dataset_1, loaded_dataset_2,
                             loaded_dataset_3, loaded_dataset_4],␣
↪ignore_index=True, sort=False)
```

```
        for index, value in dataset.iterrows():
            if value['emotion'] in ['ANG', 'angry', 'SAD', 'sad']:
                value['emotion'] = 'negative'
            elif value['emotion'] in ['HAP', 'happy', 'CAL', 'calm']:
                value['emotion'] = 'positive'
            elif value['emotion'] in ['NEU', 'neutral']:
                value['emotion'] = 'neutral'

        components = np.array([value, "Male", None, os.path.join(path, file)])
        components.append(component)

        return dataset.loc[dataset['emotion'].isin(['negative', 'positive',␣
↪'neutral'])],

        #return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

19 Load Female Quaternair Combined

[20]:
```
class QuaternairFemaleCombinedPN(Loader):
    identifier = 'combined_quaternair_Female_Positive_Negative'

    @classmethod
    def load_dataset(cls):
        components = []
        loaded_dataset_1 = NormalCrema.load_dataset()
        loaded_dataset_2 = NormalRavdess.load_dataset()
        loaded_dataset_3 = NormalSavee.load_dataset()
        loaded_dataset_4 = NormalTess.load_dataset()
        dataset = pd.concat([loaded_dataset_1, loaded_dataset_2,
                             loaded_dataset_3, loaded_dataset_4],␣
↪ignore_index=True, sort=False)

        for index, value in dataset.iterrows():
            if value['emotion'] in ['ANG', 'angry', 'SAD', 'sad']:
                value['emotion'] = 'negative'
            elif value['emotion'] in ['HAP', 'happy', 'CAL', 'calm']:
                value['emotion'] = 'positive'
            elif value['emotion'] in ['NEU', 'neutral']:
                value['emotion'] = 'neutral'

            components = np.array([value, "Female", None, os.path.join(path,␣
↪file)])
            components.append(component)
```

```
        return dataset.loc[dataset['emotion'].isin(['negative', 'positive',␣
↪'neutral'])],
        #return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

20 Load Single Record

[21]:
```
class SingleValue(Loader):
    identifier = 'single_value'

    @classmethod
    def load_dataset(cls):
        components=[]
        path = '/data/emo/notebooks/source/datasets/crema'
        for file in os.listdir(path):
            component = np.array(file.replace('.', '_').split('_'))
#             component = np.array([component[2], None, None, os.path.
↪join(path, file)])
            component = np.array(["Unknown", None, None, os.path.join(path,␣
↪file)])
            components.append(component)
            break



            if component[0] in ['ANG', 'HAP', 'SAD', 'NEU']:
                components.append(component)
                break

        print(components)
        return pd.DataFrame(data=components,␣
↪columns=['emotion','gender','subset','file_path'])
```

```
      emotion gender    subset  \
0       happy   None      Test
1     fearful   None      Test
2     fearful   None      Test
3       angry   None      Test
4       happy   None      Test
…       …     …         …
3763  fearful   None  Validate
3764    angry   None  Validate
3765  disgust   None  Validate
3766    happy   None  Validate
```

```
3767    angry   None  Validate

                                                file_path
0     /data/emo/notebooks/source/datasets/ravdess_sp…
1     /data/emo/notebooks/source/datasets/ravdess_sp…
2     /data/emo/notebooks/source/datasets/ravdess_sp…
3     /data/emo/notebooks/source/datasets/ravdess_sp…
4     /data/emo/notebooks/source/datasets/ravdess_sp…
…                                                   …
3763  /data/emo/notebooks/source/datasets/ravdess_sp…
3764  /data/emo/notebooks/source/datasets/ravdess_sp…
3765  /data/emo/notebooks/source/datasets/ravdess_sp…
3766  /data/emo/notebooks/source/datasets/ravdess_sp…
3767  /data/emo/notebooks/source/datasets/ravdess_sp…

[3768 rows x 4 columns]
```