Appendix VIII: Model Performances

In [6]:
```
"""Imports necessary packages"""

import itertools
import math
from typing import Dict, Iterable, List, Union

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import pylab
import scipy
import scipy.stats as stats
import seaborn as sns
import statsmodels.api as sm
from sklearn.model_selection import train_test_split

sns.set_style("whitegrid")
```

In [7]:
```
data = pd.read_csv("D:/School/frequentist-statistics/ITM-song-popularity/database/it
data = data.drop("Unnamed: 0", axis=1)
```

In [8]:
```
def get_mape(model_str: str, data: Iterable, epochs: int = 10, train_size: float = 0
    """Obtains mean absolute percentage error of model characterised by `model_str`

    Args:
        model_str (str): model string as required by statsmodels.formula.ols
        data (Iterable): a data set.
        epochs (int, optional): the number of iterations. Defaults to 10.
        train_size (float, optional): the relative size of the train data points as

    Returns:
        float: the mean absolute percentage error over the epochs.
    """
    mapes = []
    response = model_str.split(" ~ ")[0]
    for _ in range(epochs):
        train, test = train_test_split(data, train_size=train_size)
        model = sm.formula.ols(formula = model_str, data = train)
        model_fitted = model.fit()
        real_y = np.asarray(test[response])
        pred_y = np.asarray(model_fitted.predict(test))
        mape = []
        for i in range(len(pred_y)):
            targ = real_y[i]
            if targ == 0:
                mape.append(abs(targ - pred_y[i]))
            else:
                mape.append(abs(targ - pred_y[i])/targ)
        mapes.append(np.asarray(mape).mean())
    return np.asarray(mapes).mean()
```

In [9]:
```
models = ["popularity_abs ~ age_days + complexity + track_number", "popularity_norm
```

In [10]:
```
for model_str in models:
```

```
    print("MAPE of `%s`: %.4f" % (model_str, get_mape(model_str, data)))
```

```
MAPE of `popularity_abs ~ age_days + complexity + track_number`: 0.1524
MAPE of `popularity_norm ~ age_days + complexity + track_number`: 0.5265
MAPE of `popularity_abs ~ age_days + complexity + track_number + track_number*durati
on + danceability + duration`: 0.1406
MAPE of `popularity_norm ~ age_days + complexity + track_number + track_number*durat
ion + danceability + duration`: 0.5015
MAPE of `popularity_abs ~ track_number + duration + danceability + age_days`: 0.1580
MAPE of `popularity_norm ~ track_number + duration + danceability + age_days`: 0.544
5
MAPE of `popularity_abs ~ track_number + duration + speechiness + age_days + duratio
n*complexity + danceability*valence + danceability*complexity`: 0.1546
MAPE of `popularity_norm ~ track_number + duration + speechiness + age_days + durati
on*complexity + danceability*valence + danceability*complexity`: 0.4823
```