Project title: Make a successful song!
Members: Oliver Wood, Henri Lipping, Kaspar Matkur
REPO: https://github.com/henrilipping/Make-a-successful-song-

Task 2. Business understanding

1. Identifying your business goals

Background:

In the current music landscape, streaming has become the dominant way people listen to music, fundamentally reshaping how songs gain traction. Because listeners now have instant access to millions of tracks, the competition for attention is more intense than ever. As a result, music industry decision-makers are in a constant race to create commercially successful songs that resonate widely with audiences.

Business Goals:

Our main goal is to find a correlation between all the different song attributes and the popularity score. With that we hope that we can figure out what type of songs are most popular.

Business success criteria:

Clear analysis that identifies whether popular songs tend to share similar attribute metrics. Insights are useful for decision-makers.

2. Assessing your situation

Inventory of resources:

- 3 team members,
- 1 dataset from Kaggle (maybe more),
- Python, Jupyter Notebook,
- Python libraries: pandas, scikit-learn, matplotlib, seaborn,numpy
- 3 laptops

Requirements, assumptions, and constraints:

The data we are using is a public dataset from Kaggle, so no need for requirements.

The available data is sufficiently large and representative to train a meaningful model. The metadata for songs is complete enough (no major missing fields such as genre or popularity).

The data set may contain duplicates.

Risks and contingencies:

Technical risks like the internet or computer not working can be easily solved since we have access to multiple devices and locations to work from. Project files won't be lost as long as we use cloud-based storage systems. Training the model itself also shouldn't cause too many issues since we aren't doing anything too complicated and have gained a lot of knowledge from the course about how to use the python libraries.

Terminology

Popularity score - Spotify metric (0-100) which represents song engagement.
Audio attributes - Song descriptors (like energy, danceability, loudness, etc)

Costs and benefits

Costs: 3 people are working with the same laptops (hp elitebook 640 14 inch g11) for 30 hours. That means we work 3 x 30 = 90 hours. Our Laptops are using usually 0.045kW * 90h = 4.05kWh. Price comes then 4.05kWh * 0.23 €/kWh ≈ 0.93€.

Benefits: Insights into music trends, Ready-to-use predictive model

3. Defining your data-mining goals

Data-mining goals

We have to first look over the data and clean it if needed, then we have to find a suitable model to train and finally analyse the results and see how high accuracy we can achieve. Create graphs supporting the findings. After all that we need to prepare a poster for presentation.

Data-Mining Success Criteria

Success is achieved if the analysis reveals whether popular songs share a consistent set of attributes. Strong model performance indicates that popularity is predictable from the attributes and that meaningful patterns exist.

Task 3. Data understanding

1. **Gathering Data**

Outline data requirements

The dataset needs to contain metadata of songs, such as energy, danceability, loudness, length, etc. These variables will serve potential predictors of popularity. Secondly it also needs a popularity score assigned to each track. That is to track which song is popular or not. Thirdly it should contain additional metadata such as song names and artist names. Those may be useful for contextual understanding.

Verify data availability:

The dataset used in this project is the Spotify Tracks Attributes and Popularity dataset, obtained from Kaggle. This dataset contains approximately 114,000 rows and 21 columns, making it sufficiently large for machine learning and exploratory analysis. All necessary fields, including both audio features and the popularity score, are present and complete, confirming that the dataset is suitable for the project objectives.

Define selection criteria:

We are using a Kaggle dataset, which has 21 columns and ~114k rows. The data we use must have the necessary metadata and enough rows to be able to efficiently train a model on it. The most important attribute it must have is the popularity score, as without it, we cannot start to predict the popularity.

2. **Describing Data**

Our dataset consists of publicly available music-related data collected from Spotify. The dataset is 21 columns and 114k rows.

Useful columns:

- popularity: Spotify assigned score from 0 to 100
- artists: Artist name (for contextual understanding)
- track_name: Track name (for contextual understanding)
- track_genre: Track genre
- duration_ms: Song length in milliseconds (ms)
- explicit: Indicates whether the track contains explicit content (bool)
- tempo: Estimated beats per minute (BPM)
- valence: Musical positivity conveyed (0 = sad to 1 = happy)
- liveness: Presence of an audience in the recording (0 to 1)
- instrumentalness: Prediction on whether song has vocals or not (0 = has vocals to 1 = doesn't have vocal)
- mode: modality (major = 1, minor = 0)
- loudness: overall loudness in decibels (dB)
- energy: Intensity of the song (0 to 1)

- danceability: How suitable the track is for dancing (0 to 1)

Most of these fields are normalized between 0 and 1, except tempo, loudness, and duration. The popularity variable is numerical but will later be converted into a binary class for modeling.

3. Exploring Data

Data quality problems we found:

- Duplicate rows
- duration_ms is zero
- Empty values

Hypothesis: If we train different prediction models on different genres then we will get a more accurate model than mixing all genres into one prediction model.

To clean the dataset we should remove any duplicate rows. For rows with empty values it depends if we can retrieve the missing values and if not we remove the row.

4. Verifying Data Quality

After removing the duplicates and fixing empty values, the data is suitable for training.

Task 4. Planning your project

1. Clean dataset (remove unnecessary columns, remove dupes, handle missing values, etc) (3h, Everybody)
2. EDA (Summary stats, distributions, visualizations. Comparing popularity with other attributes using graphs to try and find any relations) (10h, Everybody)
3. Making prediction model - experimenting with different models and parameters (8h, Everybody)
4. Testing the model and finding the accuracy. Trying to improve the model. Comparing with other models. (5h, Everybody)
5. Compile findings and poster (4h, Everybody)

Tools:

- Visual studio code - IDE for coding
- Jupyter Notebook - Computing platform
- Python - Programming language
- Kaggle - Finding suitable datasets
- Docs - reports
- GitHub - version control
- Discord - for communication

Methods:

- Data cleaning
- Data exploration and visualization
- Training prediction model and evaluation
- Improving prediction model
-