OWNML MACHINE LEARNING CANVAS

Designed for: Turkish Data Base   Designed by: Team 18   Date: 12/Oct/2025   Iteration: 5

PREDICTION TASK

1. **Type of task**: Supervised multiclass classification with 4 emotion categories (happy, sad, angry, relax).
2. **Prediction entity:** Each prediction is made on a 30-second audio clip and then aggregated at the track level.
3. **Outputs:** The model provides probability values for each emotion and assigns the final predicted label.
4. **Observation timing:**
   - Training: Ground-truth emotion labels are available offline.
   - Production: User interactions such as “likes” and “skips” are collected to monitor and validate performance.

DECISIONS

1. **Trigger:** A decision is executed every time the system selects the next song during playback.
2. **Inputs**: The decision uses the emotion probabilities of each track and the user’s selected target mood.
3. **Decision rule**:
   - If the confidence score for a track ≥ threshold value (for example, 0.6), the system plays that track.
   - If the confidence score < threshold, the system switches to the standard shuffle mode.
4. **Latency requirement**: The complete decision process must be finished in less than 200 milliseconds.
5. **Logging**: Each decision records the target mood, the selected track identification, and the confidence level for later evaluation and A/B testing.

VALUE PROPOSITION

1. **End beneficiaries:** Streaming platform users and internal product teams.
2. **Pain points:**
   - The normal shuffle mode ignores user mood, which decreases engagement by 10–20%.
   - Manual playlist curation is inconsistent and time-consuming.
3. **Solution**: The “Smart Shuffle by Mood” feature automatically selects songs that match the user’s emotional preference.
4. **Benefits**: The system increases personalization, improves user satisfaction, and reduces skipped tracks.
5. **Integration:**
   - Implemented in the player interface as a mood selection option.
   - Connected through an API to the recommendation engine.
   - Performance and engagement tracked through analytic dashboards.

DATA COLLECTION

1. **Initial sourcing**: Import audio clips and labels, verify that each has a valid duration and format, and store for processing.
2. **Feature generation**: Compute chroma and spectrogram features using fixed parameters; split the dataset into training, validation, and testing.
3. **Update frequency:**
   - New tracks are processed daily or in near real time.
   - User feedback data is aggregated every hour to adjust threshold values.
4. **Quality assurance**: Validate for correct duration, balanced classes, and acceptable signal-to-noise ratio (SNR > minimum limit).
5. **Cost management**: Use batch processing, cached features, and automatic scaling to reduce computational cost.

DATA SOURCES

1. **Audio clips**: 400 audio files, each 30 seconds long, stored in WAV or MP3 format.
2. **Emotion labels:** One label per clip, balanced across the four emotion classes (100 per class).
3. **Derived features**: Chroma and spectrogram tensors generated during preprocessing.
4. **User feedback**: Play, skip (<30 seconds), like, and completion events collected through APIs or platform logs.

IMPACT SIMULATION

1. **Positive impact**: Correct predictions increase listening time by approximately 3-8% and reduce skipped songs by 10-20%.
2. **Negative impact:** Incorrect predictions decrease average session time by 5-10%.
3. **Simulation data**: Historical playback logs and user feedback are analyzed before deployment to predict performance.
4. **Deployment criteria:**
   - Accuracy ≥85%.
   - Engagement improvement ≥3% in A/B testing.
   - Latency ≤200 milliseconds.
5. **Fairness**: Accuracy difference across moods and genres must remain <5%.

MAKING PREDICTIONS

1. **Prediction mode**: Real-time inference during song playback.
2. **Frequency:** One prediction made each time a new song is selected.
3. **Response time**: The prediction and decision process must finish in ≤200 milliseconds.
4. **Infrastructure:** Runs on lightweight GPU or CPU servers in cloud or edge environments, using cached embeddings to ensure fast processing.

BUILDING MODELS

1. **Number of models**: One main classifier for emotion prediction.
2. **Retraining policy**: Model retrained every month or earlier if accuracy decreases by >5% or if new labeled data becomes available.
3. **Training duration**: Each training and evaluation cycle takes between 2 and 4 hours.
4. **Computational resources**: GPU-enabled environment (for example, NVIDIA T4 or A100) using TensorFlow or PyTorch, tracked with DVC and MLflow for reproducibility.

FEATURES

5. **Representation:** Each 30-second segment is converted into a chroma spectrogram (224 × 224 pixels) for CNN input.
6. **Preprocessing:** Resample audio at 22.05 kHz, normalize loudness, apply log scaling, and standardize using z-score.
7. **Feature extraction**: Deep embeddings (4096-dimensional) are extracted from the fully connected layer (fc7) of a pre-trained VGG-16 model.
8. **Aggregation**: Average all probability values across windows to generate one final emotion label per track.
9. **Storage**: Features are stored in a dedicated directory for reuse during inference.

MONITORING

1. **Model performance metrics:** Macro-F1, accuracy, latency ≤200 milliseconds (95th percentile), and detection of drift in input features or predictions.
2. **Business indicators:** Average listening time, skip rate, and user satisfaction measured through likes or survey results.
3. **Review frequency**:
   - Daily: automated operational dashboard.
   - Weekly: model and KPI review.
   - Retraining if accuracy decreases by >5% or if drift is detected.