Task 1 – Video Game Reviews

The rise of online platforms has caused a huge increase in user-created content, like video game reviews. These reviews usually include short written summaries and a score from 1 to 10. While it's easy to collect the written reviews, getting accurate human-assigned scores for each one takes a lot of time and effort. This project focuses on predicting review scores when there aren't many labeled examples available. It looks at different machine learning methods, including semi-supervised learning (SSL). The goal is to build accurate prediction models that use a small amount of labeled data along with a large amount of unlabeled data, reducing the need for lots of manual work.

Dataset Description

Given a collection of video game review summaries, some with associated numerical scores (1-10) and a larger set without, the task is to accurately predict the numerical score for unseen review summaries. The primary challenge lies in the scarcity of labeled data. The project utilizes two distinct datasets, simulating a realistic scenario with limited annotation budget:

- Labeled Dataset (labeled_reviews.csv): Contains two columns: review_text (textual review content) and review_score (integer,1-10).This dataset represents the scarce, high-quality human-annotated data available for initial model training and validation. Its size is deliberately constrained to simulate real-world limitations in data annotation.
- Unlabeled Dataset (unlabeled_reviews.csv): Contains a single column: review_text (textual review content).This larger pool of data is available without associated scores and will be leveraged by semi-supervised and active learning techniques to enhance model generalization without incurring additional manual labeling costs.

The deliberate imbalance in dataset sizes (small labeled, large unlabeled) is central to evaluating the efficacy of techniques designed for low-resource environments.

1. Text Vectorization (21 Points)

Raw textual data must be transformed into numerical representations suitable for machine learning algorithms. These transformations aim to capture the semantic and syntactic properties of the review summaries.

- **SentenceTransformer (Semantic Embeddings):**
  - A neural network model designed to produce dense vector embeddings for entire sentences or paragraphs. These embeddings are optimized such that semantically similar texts are mapped to proximate points in the vector space.

    This makes them highly effective for tasks requiring a deep understanding of sentence meaning.
  - Implementation (7 Points):
    - Install the sentence-transformers library.
    - Load a pre-trained model (all-MiniLM-L6-v2) suitable for general-purpose sentence embeddings.
    - Compute embeddings for all summaries in both the labeled and unlabeled datasets.
- **Word2Vec (Distributed Word Representations):**
  - A predictive model that learns continuous vector representations for words. It operates by predicting surrounding words given a target word or predicting a target word given its context . Word vectors capture semantic relationships (e.g., "king" - "man" + "woman" = "queen"). Sentence-level representations can be derived by aggregating the vectors of all words within a summary.
  - Implementation(7 Points):
    - Utilize the Gensim library to train a Word2Vec model on the combined corpus of all review summaries (both labeled and unlabeled).
    - After training, compute sentence embeddings for each summary by averaging the vectors of its constituent words.
- **Dimensionality Reduction and Visualization:**
  - High-dimensional embeddings can be difficult to visualize. Techniques like Principal Component Analysis (PCA) help reduce the number of dimensions while preserving as much of the original variance as possible. This makes it easier to visually inspect clusters and patterns.
  - Implementation (7 Points):
    - Perform PCA on the generated embeddings to reduce their dimensionality, and create a scatter plot of the resulting data points. Use color to represent the actual scores (for labeled data) to visualize potential clustering and patterns in the reduced space.
    - If you don't observe any clear pattern or clustering in the scatter plot, explain why that might be the case.

2. Supervised Learning Baselines (17 Points)

Before employing advanced techniques, establish baseline performance using only the labeled data. The nature of the score (discrete integers 1-10) allows for two primary modeling paradigms:

- **Classification Paradigm:**
  - Treats each score (1 through 10) as a distinct categorical class. The model learns to assign a specific class label to each review summary.

    - Considerations: Directly handles discrete outputs but might not explicitly leverage the ordinal relationship between scores (e.g., a score of 8 is "closer" to 7 than to 1).
- **Regression Paradigm:**
    - Treats the score as a continuous numerical value. The model aims to predict a real-valued score, which can then be rounded to the nearest integer for final prediction.
    - Considerations: Captures the ordinal nature of scores and allows for predictions between integer values. However, it might produce predictions outside the valid range (1-10) if not constrained.
- **Implementation (17 Point):**
    - Data Split: Partition the labeled dataset into training (80%), validation (10%), and testing (10%) sets.
    - Model Training:
        - Train a suitable classifier (e.g., Logistic Regression, Random Forest Classifier, Support Vector Classifier).
        - Train a suitable regressor (e.g., Linear Regression, Support Vector Regressor, Random Forest Regressor).
    - Evaluation:
        - Classification Metrics: Accuracy, Precision, Recall, F1-score (macro-averaged for multi-class), and Confusion Matrix.
        - Regression Metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R2).
    - Comparative Analysis: Evaluate and compare the performance of classification and regression models to identify which approach is more effective for this task. Select the model that demonstrates superior performance for further use.

3. Semi-Supervised Learning (SSL) Strategies (50 Points)

SSL techniques aim to improve model performance by leveraging the large pool of unlabeled data in conjunction with the small labeled set.

3.1. Pseudo-Labeling

- An iterative SSL approach where a model, initially trained on the small labeled dataset, is used to generate "pseudo-labels" for a subset of the unlabeled data. Only predictions made with high confidence are selected. These pseudo-labeled samples are then added to the training set, and the model is retrained on the expanded dataset. This process can be repeated.
- **Key Concepts:**

    1. Confidence Threshold: A critical hyperparameter defining the minimum prediction probability (for classification) or maximum prediction uncertainty (for regression) required for a pseudo-label to be accepted.
    2. Iterative Refinement: The process can be repeated over multiple rounds, potentially adding more pseudo-labeled samples and refining the model's performance.
    3. Risk of Confirmation Bias: If the initial model makes systematic errors, these errors can be reinforced by adding incorrect pseudo-labels, potentially leading to performance degradation. Careful threshold selection is crucial.
- **Implementation (25 Point):**
    1. Train the best-performing baseline model (classifier or regressor) on the initial labeled training set.
    2. Predict scores for all samples in the unlabeled dataset.
    3. Select unlabeled samples for which the model's prediction confidence exceeds a predefined threshold.
    4. Assign these high-confidence predictions as pseudo-labels to the selected unlabeled samples.
    5. Combine the original labeled training data with the newly pseudo-labeled data.
    6. Retrain the model on this expanded dataset.
    7. Evaluate the retrained model on the held-out test set.
    8. If applying this iteratively improves model performance, repeat steps 2–7 for multiple rounds, observing performance changes.

3.2. Active Learning

Active learning is a machine learning paradigm where the learning algorithm interactively queries a human oracle (annotator) to label new data points. The goal is to strategically select the most "informative" unlabeled samples for labeling, thereby maximizing model improvement with minimal annotation effort.

- **Common Query Strategies:**
    1. Least Confidence Sampling: Selects the unlabeled sample for which the model has the lowest prediction probability for its most likely class (for classification) or highest prediction uncertainty (for regression).
    2. Margin Sampling: Selects samples where the difference between the probabilities of the top two predicted classes is smallest, indicating high uncertainty between competing classes.
    3. Entropy-Based Sampling: Selects samples with the highest predictive entropy, which quantifies the overall uncertainty across all possible class predictions.
- **Implementation (25 Point):**
    1. Start with the initial small labeled training set and train the chosen baseline model.

2. For each active learning round:
    - Compute uncertainty scores (based on chosen strategy) for all samples in the unlabeled pool.
    - Select the top-k most uncertain unlabeled samples.
    - Simulate human annotation: For this project, you will manually assign a score to these selected k samples.
    - Add these newly labeled samples to the training set.
    - Retrain the model on the expanded labeled dataset.
    - Record the model's performance on the test set.
3. Repeat for a predetermined number of rounds (e.g., 5-10 rounds), plotting model performance against the cumulative number of labeled examples.

4. Comparative Performance Analysis (22 Points)

- **Summary of Metrics (5 Point):**
  Tabulate the key evaluation metrics (accuracy, F1-score, MAE, RMSE, etc.) for:
    - The initial baseline model (trained only on the small labeled set).
    - The model after each round of Pseudo-Labeling.
    - The model after each round of Active Learning.
- **ROC and AUC Curves (7 Point):**
  Plot ROC (Receiver Operating Characteristic) curves and compute AUC (Area Under the Curve) scores for each of the above models to assess their ability to distinguish between classes.
- **Learning Curves (5 Point):**
  Plot learning curves showing model performance (e.g., F1-score or MAE) on the test set against the total number of labeled samples (original + pseudo-labeled + actively queried).
- **Discussion (5 Point):**
  Analyze and discuss the effectiveness of Pseudo-Labeling and Active Learning. Which method yielded the most significant performance improvement? Under what circumstances (e.g., specific dataset characteristics, model type) might one method be preferred over the other? Discuss the trade-offs and potential pitfalls of each SSL approach, such as the risk of confirmation bias in Pseudo-Labeling or high labeling costs in Active Learning.