## Project Documentation: Movie Review Sentiment Analysis

### Project Title: Mini Deep Learning Project - Sentiment Analysis of Movie Reviews

### 1. Business Objective

The purpose of this project is to \*\*enhance customer insights through sentiment analysis of movie reviews\*\*, enabling studios and marketing teams to gain a deeper understanding of audience sentiment. This insight can guide marketing strategies and production decisions, tailoring movie promotion efforts and even influencing content creation based on audience preferences.

#### Business Constraints

1. \*\*Data Quality\*\*: Ensure that the dataset is well-prepared, with representative samples of diverse viewer opinions to avoid sentiment bias.

2. \*\*Computational Resources\*\*: Consider the limitations of available computational resources, particularly when training deep learning models on larger datasets.

3. \*\*Interpretability\*\*: The model's sentiment predictions should be transparent and understandable for stakeholders unfamiliar with machine learning.

#### Success Criteria

1. \*\*Business Success\*\*: Achieve an accuracy of at least 85% in classifying movie reviews as positive or negative, providing meaningful insights for marketing and production decision-making.

2. \*\*Model Performance\*\*: During training, the model should achieve a loss below 0.3 and a validation accuracy of at least 90%.

3. \*\*Outcome Reporting\*\*: The results should be presented through detailed written reports, interpreting findings and explaining model performance metrics, without relying on visual aids.

### 2. Data Collection

For this project, we used the \*\*IMDb dataset\*\*, a widely used dataset containing thousands of labeled movie reviews for sentiment analysis tasks. To streamline the model training and improve performance, only the top 10,000 most common words in the dataset were retained, ensuring the focus on frequently used terms and reducing vocabulary size.

### 3. Project Workflow

#### Step 1: Import Necessary Libraries

To implement the deep learning model, the following libraries were imported:

- \*\*TensorFlow\*\*: For building and training the deep learning model.

- \*\*Sequential\*\*: To set up a layer-by-layer structure in the neural network.

#### Step 2: Load and Explore the IMDb Dataset

The IMDb dataset was loaded using TensorFlow's built-in data loader, retaining only the 10,000 most common words in reviews. This step ensures a manageable vocabulary size, focusing on key terms most indicative of sentiment. The first few rows of the data were printed to validate the data structure.

#### Step 3: Model Initialization and Architecture Design

A \*\*Sequential\*\* model was initialized for the sentiment analysis task. The model architecture includes:

1. \*\*Embedding Layer\*\*: Converts each word into dense vectors, enabling the model to learn word relationships.

2. \*\*Flattening Layer\*\*: Transforms the 2D word embeddings into a 1D feature vector to feed into the dense layers.

3. \*\*Fully Connected (Dense) Layers\*\*: Adds fully connected layers with ReLU activation for hidden layers and sigmoid activation for the output layer, enabling binary classification for positive or negative sentiment.

#### Step 4: Model Compilation

The model was compiled using the following configurations:

- \*\*Optimizer\*\*: Adam, chosen for its adaptive learning rate, accelerating convergence.

- \*\*Loss Function\*\*: Binary Cross-Entropy, ideal for binary classification tasks.

- \*\*Metrics\*\*: Accuracy, allowing easy tracking of the model's performance over training epochs.

#### Step 5: Logging for Non-Visual Reporting

A \*\*CSVLogger\*\* was initialized to save training logs in CSV format. This enables data tracking without visualizations, aligning with the project’s reporting requirements.

#### Step 6: Model Training

The model was trained on the training dataset using the following parameters:

- \*\*Epochs\*\*: 10

- \*\*Batch Size\*\*: 64

- \*\*Validation Split\*\*: 20% (0.2), allowing for real-time validation of the model’s performance after each epoch.

The model's training progress was monitored through accuracy and loss metrics, which were saved via the CSVLogger for later analysis.

#### Step 7: Model Evaluation

Post-training, the model was evaluated on the test data. Key performance metrics, such as accuracy and loss, were recorded. The focus was on achieving the target validation accuracy of 90% while keeping the loss below 0.3, ensuring that the model met the defined success criteria.

#### Step 8: Interpretation of Results

Using the logged data from CSVLogger, the model’s performance was analyzed through non-visual reporting techniques:

- \*\*Accuracy and Loss Summary\*\*: Provided written analysis of training and testing accuracy and loss, detailing whether the model met business and performance criteria.

- \*\*Metric-Based Insights\*\*: Presented insights based on evaluation metrics, offering interpretations relevant to stakeholders for practical decision-making.

### 4. Summary and Insights

This project established a deep learning model for movie review sentiment analysis, aiming to assist decision-makers in understanding audience sentiment with 85% or higher accuracy. The model’s architecture and training process were designed to optimize performance within the computational constraints and to provide interpretable results. The logged data allowed for detailed reporting of findings, ensuring that the sentiment analysis insights are accessible and actionable for marketing and production teams.