**Model Optimization and Tuning Phase Template**

|  |  |
| --- | --- |
| Date | 26 June 2025 |
| Team ID | SWTID1750052396 |
| Project Title | Analysis of Medium App Reviews from Google Play Store |
| Maximum Marks | 10 Marks |

**Model Optimization and Tuning Phase**

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

### Hyperparameter Tuning Documentation (8 Marks):

|  |  |
| --- | --- |
| **Model** | **Tuned Hyperparameters** |
| **Model 1: LSTM** | - **Embedding Dim = 128**: Size of the word embeddings.  - **LSTM Units = 64**: Number of units in the LSTM layer.  - **Dropout = 0.3**: Regularization to prevent overfitting.  – **Batch Size = 32**: Number of samples per batch during training.  - **Epochs = 10**: Number of training iterations.  - **Learning Rate = 0.001** (Adam Optimizer).  #code snippet  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout  from tensorflow.keras.optimizers import Adam  model\_lstm = Sequential()  model\_lstm.add(Embedding(input\_dim=5000, output\_dim=128, input\_length=200))  model\_lstm.add(LSTM(units=64))  model\_lstm.add(Dropout(0.3))  model\_lstm.add(Dense(1, activation='sigmoid'))  optimizer = Adam(learning\_rate=0.001)  model\_lstm.compile(loss='binary\_crossentropy', optimizer=optimizer, metrics=['accuracy'])  # Training  history\_lstm = model\_lstm.fit(X\_train, y\_train, batch\_size=32, epochs=10, validation\_split=0.2) |
| **Model 2: BiLSTM** | - **Embedding Dim = 128**: Same as above, but better captures context.  - **BiLSTM Units = 64**: Processes text in both directions.  - **Dropout = 0.4**: Increased to reduce overfitting.  - **Batch Size = 64**: Faster convergence with more data per step.  - **Epochs = 12**: Allowed for more learning. - **Learning Rate = 0.0005** (Adam with decay).  #code snippet  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Embedding, Bidirectional, LSTM, Dense, Dropout  from tensorflow.keras.optimizers import Adam  model\_bilstm = Sequential()  model\_bilstm.add(Embedding(input\_dim=5000, output\_dim=128, input\_length=200))  model\_bilstm.add(Bidirectional(LSTM(units=64)))  model\_bilstm.add(Dropout(0.4))  model\_bilstm.add(Dense(1, activation='sigmoid'))  optimizer = Adam(learning\_rate=0.0005)  model\_bilstm.compile(loss='binary\_crossentropy', optimizer=optimizer, metrics=['accuracy'])  # Training  history\_bilstm = model\_bilstm.fit(X\_train, y\_train, batch\_size=64, epochs=12, validation\_split=0.2) |
|  |  |

### 

### Final Model Selection Justification (2 Marks):

|  |  |
| --- | --- |
| **Final Model** | **Reasoning** |
| **Model 2: BiLSTM** | Model 2 (BiLSTM) was selected as the final optimized model because it outperformed the baseline LSTM in terms of validation accuracy, F1-score, and generalization. The bidirectional architecture allowed the model to capture both past and future context in user reviews, leading to more accurate sentiment predictions. It also demonstrated better stability across epochs with reduced overfitting due to tuned dropout and learning rate. |