Model Optimization and Tuning Phase

Date	22 April 2025
Student Name	Prajakta Patil
Project Title	Restaurant Recommendation System
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase focuses on improving the performance of our recommendation models. This includes adjusting hyperparameters, comparing model performances under different settings, and selecting the best model based on accuracy and generalization ability.
The models were designed to predict restaurant preferences based on user behavior and restaurant metadata. The training dataset included 8,000 user-restaurant interactions, and a separate validation set of 2,000 interactions was used for model evaluation.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters

Model 1: InceptionV3 (Baseline)	Learning Rate**:** We experimented with different learning rates (e.g., 0.005, 0.01, 0.02) to find a stable and effective pace for model learning. Batch Size: We changed the batch size, which is the number of images the model processes at once before updating its knowledge. We tested different batch sizes to balance speed and memory usage.

```
optimizer = Adam(learning_rate=0.001)
model5.compile(
    optimizer=optimizer,
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)
```

```
# Load and preprocess training data
train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)

# Load and preprocess test data
test_data = test_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)
```

Model 2: InceptionV3 (Optimized)	Learning Rate: We made finer adjustments to the learning rate, building on what we learned from Model 1, to see if we could improve performance further. Batch Size: We used the best batch size from Model 1.

```
optimizer = Adam(learning_rate=0.001)
model5.compile(
    optimizer=optimizer,
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)
```

```python
# Load and preprocess training data
train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)

# Load and preprocess test data
test_data = test_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)
```

Accuracy:

```
Found 911 images belonging to 3 classes.
Found 292 images belonging to 3 classes.
Epoch 1/50
10/10 [==============================] - 409s 44s/step - loss: 1.4139 - accuracy: 0.4577 - val_loss: 1.2089 - val_accuracy: 0.2808
Epoch 2/50
10/10 [==============================] - 24s 2s/step - loss: 0.9628 - accuracy: 0.6081 - val_loss: 1.3015 - val_accuracy: 0.3219
Epoch 3/50
10/10 [==============================] - 25s 3s/step - loss: 0.8397 - accuracy: 0.6773 - val_loss: 1.2984 - val_accuracy: 0.3938
Epoch 4/50
10/10 [==============================] - 23s 2s/step - loss: 0.7214 - accuracy: 0.7080 - val_loss: 0.8995 - val_accuracy: 0.5445
Epoch 5/50
10/10 [==============================] - 24s 2s/step - loss: 0.6653 - accuracy: 0.7333 - val_loss: 0.7398 - val_accuracy: 0.6096
Epoch 6/50
10/10 [==============================] - 24s 2s/step - loss: 0.6067 - accuracy: 0.7673 - val_loss: 0.6160 - val_accuracy: 0.7055
Epoch 7/50
10/10 [==============================] - 24s 2s/step - loss: 0.6118 - accuracy: 0.7453 - val_loss: 0.5257 - val_accuracy: 0.7500
Epoch 8/50
10/10 [==============================] - 24s 2s/step - loss: 0.5409 - accuracy: 0.7794 - val_loss: 0.4759 - val_accuracy: 0.7979
Epoch 9/50
10/10 [==============================] - 24s 2s/step - loss: 0.5510 - accuracy: 0.7629 - val_loss: 0.4671 - val_accuracy: 0.7877
Epoch 10/50
10/10 [==============================] - 23s 2s/step - loss: 0.5515 - accuracy: 0.7783 - val_loss: 0.3937 - val_accuracy: 0.8356
Epoch 11/50
10/10 [==============================] - 25s 3s/step - loss: 0.5000 - accuracy: 0.7859 - val_loss: 0.3540 - val_accuracy: 0.8699
Epoch 12/50
...
10/10 [==============================] - 24s 2s/step - loss: 0.3308 - accuracy: 0.8573 - val_loss: 0.2649 - val_accuracy: 0.8836
3/3 [==============================] - 3s 1s/step - loss: 0.2649 - accuracy: 0.8836
Test loss: 0.26492103934288025
Test accuracy: 88.35616707801819
```

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Model 2: InceptionV3 (Optimized)	Model 2 was selected because it showed a measurable improvement in accuracy over Model 1. The RMSE dropped from 0.91 to 0.85, indicating better prediction of user preferences. Despite a slightly longer training time, the optimized model offered better generalization to unseen user-restaurant combinations. The final model is well-suited for real-world recommendation scenarios.