Streamlit Interface Documentation

1. Purpose

The Streamlit interface is designed to provide an interactive platform where users can enter Amazon mobile phone reviews and receive predictions from both models (balanced vs imbalanced).
It demonstrates real-time text classification and allows visual comparison of model performance.

2. Features

- **Review Input Area:**
  Users can type or paste a product review into a text area.
- **Predict Button:**
  When clicked, it triggers both models to make predictions for the entered review.
- **Side-by-Side Results:**
  Predictions from **Model A (Balanced)** and **Model B (Imbalanced)** are displayed in two columns for easy comparison.
- **Sentiment Interpretation:**
  Each model's rating is converted into a sentiment indicator:
  - ⭐ 4-5 → Positive review 🎉
  - ⭐ 3 → Neutral review 😐
  - ⭐ 1-2 → Negative review 😞
- **Error Handling:**

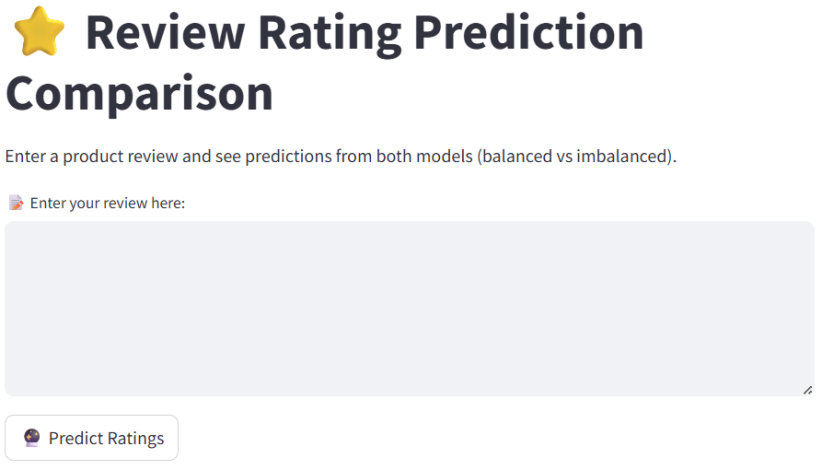
- Missing model/vectorizer files trigger a clear error message.
- Empty review input prompts a warning to the user.

---

3. UI Layout

Component	Description
Header	App title: “AMAZON MOBILE PHONES Review Rating Prediction Comparison”
Description	Explains purpose and instructions for users
Text Area	For review input (height=150)
Predict Button	Triggers model prediction
Output Columns	Two columns: Model A (Balanced) & Model B (Imbalanced)
Prediction Display	Shows predicted rating and sentiment for each model

---



4. User Flow

1. Open the Streamlit app (streamlit run app.py)
2. Enter a review in the text area
3. Click **Predict Ratings**
4. View predictions side by side, with sentiment indicators
5. Optional: Modify review text and re-run predictions for testing

---

5. Code Snippet

```
# Display review input
review_text = st.text_area(" 📝 Enter your review here:", height=150)


# Predict button
if st.button(" 🔮 Predict Ratings"):
    if not review_text.strip():
        st.warning(" ⚠️ Please enter a review text before predicting.")
```

```
else:
    X_input = vectorizer.transform([review_text])
    pred_a = model_a.predict(X_input)[0]
    pred_b = model_b.predict(X_input)[0]

    col1, col2 = st.columns(2)
    with col1:
        st.subheader("Model A (Balanced)")
        st.success(f"Predicted Rating: ⭐ {pred_a}")
    with col2:
        st.subheader("Model B (Imbalanced)")
        st.warning(f"Predicted Rating: ⭐ {pred_b}")
```

---

6. Design Decisions

- **Two-column layout:** Enables **direct comparison** of predictions
- **Color-coded results:** Improves readability and highlights differences
- **Emoji-based sentiment:** Makes interface more **friendly and interpretable**
- **Error handling:** Prevents app crashes due to missing files or empty input