Bonus Task: Predicting whether or not a meme is toxic

Building upon the existing LLaVa model implementation as shown in approach 2 of task 1, we can extend it to tackle the problem of toxicity classification based on meme text. The relevant information from the previous documentation has been included, and new components specific to the toxicity classification task have been added.

Toxicity Classification using LLaVa

1. **Model Initialization**
2. **Textual Data Processing**
3. **Toxicity Classification Model**
   - Implement or utilize a separate toxicity classification model trained specifically on meme text and corresponding toxicity labels.
   - This model should be capable of predicting the toxicity level of the generated meme text descriptions.
   - Potential model architectures include transformer-based models like BERT, RoBERTa, or XLNet, fine-tuned on meme text toxicity datasets.
4. **Toxicity Prediction**
   - Pass the textual descriptions generated by LLaVa to the toxicity classification model.
   - Obtain the predicted toxicity level or score for each meme text.
5. **Implementation Details**
   - Extend the existing `ImageDescriptionGenerator` class with a new method to handle toxicity classification.
   - This new method should take the image path as input, generate the textual description using LLaVa, and pass it through the toxicity classification model.
   - The method should return the predicted toxicity level or score.

```
class ToxicityClassifier:
    def __init__(self, model_id, toxicity_model_path):
        self.image_description_generator = ImageDescriptionGenerator(model_id)
        self.toxicity_model = load_toxicity_model(toxicity_model_path)

    def classify_toxicity(self, image_path):
        description = self.image_description_generator.generate_description(image_path)
        toxicity_score = self.toxicity_model.predict(description)
        return toxicity_score
```

The `ToxicityClassifier` class encapsulates both the LLaVa model for generating textual descriptions and the toxicity classification model. The `classify_toxicity` method orchestrates the entire process, taking an image path as input and returning the predicted toxicity score.

Potential Enhancements and Future Work

- This section remains the same as described in the previous documentation.

By integrating the toxicity classification model with the existing LLaVa implementation, we can leverage the multimodal capabilities of LLaVa to generate context-aware textual descriptions, which can then be fed into the toxicity classification model for accurate and efficient toxicity prediction based on meme text.