Movie Recommendation System Report

1. Introduction
A hybrid recommendation system combining:
- **Natural Language Processing (NLP)** for content analysis
- **Machine Learning (ML)** for similarity matching
- **Reinforcement Learning (RL)** for strategy optimization

2. Technical Components

2.1 NLP Pipeline
```python
# Text preprocessing example
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    words = nltk.word_tokenize(text)
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return ' '.join(words)
```

2.2 Machine Learning
- TF-IDF Vectorization
- Cosine Similarity
- Three recommendation strategies:
  1. Content-based
  2. Genre-based
  3. Popularity-based

2.3 Reinforcement Learning
- Custom Gym environment
- DQN (Deep Q-Network) agent
- Reward function based on simulated user feedback

3. Implementation

System Workflow
1. User requests recommendations
2. NLP processes the query
3. RL agent selects best strategy
4. System returns recommendations
5. Feedback improves future suggestions

4. Results

Sample Recommendation Output:

```
Recommended Movies (content-based):
 - The Dark Knight (2008) | Action, Thriller | ⭐4.7
 - Inception (2010) | Action, Sci-Fi | ⭐4.8
 - Interstellar (2014) | Adventure, Sci-Fi | ⭐4.6
```

5. How to Run

1. Install dependencies:
```bash
pip install -r requirements.txt
```

2. Launch Jupyter notebook:
```bash
jupyter notebook code/movie_recommender.ipynb
```

6. Future Improvements
- Add real user feedback collection
- Incorporate collaborative filtering
- Deploy as web application

References
[1] Reinforcement Learning for Recommender Systems (2022)
[2] NLP Techniques for Content-Based Filtering (2021)