**Project Report – MOVIE RECOMMENDATION SYSTEM**

|  |  |
| --- | --- |
| **GROUP NAME :** | GCEE-SEABORN |

|  |  |  |  |
| --- | --- | --- | --- |
| **GROUP MEMBERS:** | 1. 21IMT03 | - | Arun Prakash G |
|  | 1. 21IMT04 | - | Ashwanthjee A R |
|  | 1. 21IMT05 | - | Ayyanar S |
|  | 1. 21IMT64L | - | Rutchagan G |
|  | 1. 21IMT38 | - | Shobika T |

1. **ABSTRACT :**

This project aims to create a web application that recommends movies based on user emotions. The application employs machine learning techniques to predict the genre of movies corresponding to an emotion inputted by the user. Upon receiving the emotion, the app predicts the genre and filters a list of movies to provide top-rated suggestions. The project utilizes Flask for the web framework, sklearn for machine learning, and matplotlib for visualizations. Additionally, it includes mechanisms for loading and filtering movie data from a CSV file, ensuring efficient data handling and presentation.

1. **MODEL IMPLEMENTATION :**

The model implementation begins with the preparation of training data, mapping emotions to genres. A TF-IDF vectorizer converts these emotions into numerical features. The Logistic Regression model is trained using these features to predict the genre. This model is serialized and saved to disk for later use in making predictions. The vectorizer and the model, along with a genre encoder, are encapsulated in a pickle file to facilitate easy loading and prediction during runtime.

1. **ALGORITHM IMPLEMENTATION :**

**Training Data Preparation**

* **Emotion-Genre Mapping**: A dictionary maps various emotions to corresponding movie genres.
* **Vectorization**: Emotions are converted into numerical data using TF-IDF vectorization.
* **Encoding**: Genres are encoded into numerical labels for training purposes.

**Model Training**

* **Logistic Regression**: The Logistic Regression algorithm is trained on the vectorized emotion data and encoded genres.
* **Model Serialization**: The trained model and vectorizer are saved using pickle.

**Model Prediction**

* **Loading Model**: The model and vectorizer are loaded from the pickle file.
* **Prediction**: The input emotion is vectorized, and the model predicts the genre by mapping the predicted label back to the genre.

1. **PREDICATION COMPARISON REPORT :**

In the visualizations.py script, two machine learning models, Logistic Regression and Support Vector Machine (SVM), are trained and evaluated on the movie dataset to compare their performance in predicting movie genres based on text data.

**Accuracy Scores**

* **Logistic Regression**: Achieved an accuracy of 0.95.
* **SVM**: Achieved an accuracy of 0.97.

**Visualization of Results**

* **Genre Distribution**: A bar plot visualizes the distribution of genres within the movie dataset, highlighting the most and least common genres.
* **Ratings Distribution**: A histogram shows the distribution of movie ratings, providing insights into the spread of ratings in the dataset.
* **Model Accuracy Comparison**: A bar plot compares the accuracies of Logistic Regression and SVM models, indicating the effectiveness of each model in predicting genres.

1. **FINAL PREDICTION :**

When a user inputs an emotion through the web interface, the system:

1. **Predicts Genre**: The predict\_genre function uses the Logistic Regression model to predict the most suitable movie genre based on the provided emotion.
2. **Loads Movies**: The load\_movies\_from\_csv function loads the list of movies from a CSV file.
3. **Filters Movies**: The filter\_movies\_by\_genre function filters the loaded movies to include only those that match the predicted genre.
4. **Sorts and Presents Movies**: The filtered movies are sorted by rating and the top 70 movies are presented to the user.
5. **CONCLUSION :**

The project successfully demonstrates the integration of machine learning into a web application to enhance user experience through emotion-based movie recommendations. The Logistic Regression model, with its high accuracy, effectively maps user emotions to movie genres, ensuring relevant recommendations. The inclusion of visualizations aids in understanding the dataset and model performance, while the Flask framework provides a seamless interface for user interaction. The system's ability to accurately predict genres and recommend top-rated movies makes it a valuable tool for users seeking personalized movie suggestions. Future improvements could involve expanding the emotion-genre mapping and incorporating more sophisticated models to further enhance prediction accuracy.