

## Diagrama de classes simples

+-----+

|    DataLoader    |

+-----+

| - batch\_size: int |

| - shuffle: bool   |

+-----+

| + create\_data\_loader(X, y): DataLoader |

+-----+

+-----+

|    Model    |

+-----+

|            |

+-----+

| - vocab\_size: int |

| - num\_genres: int |

| - model: Sequential|

| - criterion: BCELoss|

| - optimizer: Adam   |

+-----+

| + train\_model(loader: DataLoader, num\_epochs: int): void |

+-----+

+-----+

|    DataUtils    |

+-----+

```
| + load_data(train_file: str, test_file: str): tuple |
| + preprocess_data(train_data, test_data): tuple |
| + save_accuracy_to_csv(filename: str, accuracy_values: list): void |
```

```
+-----+
```

```
+-----+
```

```
|   CountVectorizer   |
```

```
+-----+
```

```
| - max_features: int |
```

```
+-----+
```

```
+-----+
```

```
|   torch.nn.Module   |
```

```
+-----+
```

```
| + forward(inputs: Tensor): Tensor |
```

```
+-----+
```

```
+-----+
```

```
|   DataLoader         |
```

```
+-----+
```

```
| + __init__(data: TensorDataset, batch_size: int, shuffle: bool): void |
```

```
+-----+
```

```
+-----+
```

```
|   TensorDataset     |
```

```
+-----+
```

```
| + __init__(X: Tensor, y: Tensor): void |
```

```
|   torch.Tensor      |
```

```
+-----+
| - dtype: type |
```

### Explicações

DataLoader, Model, e DataUtils são classes que encapsulam funcionalidades específicas.

CountVectorizer e torch.nn.Module são classes que são utilizadas internamente.

TensorDataset, DataLoader, e torch.Tensor são classes relacionadas à manipulação de dados e tensores no PyTorch.

## Diagrama de casos de uso

```
|  Usuário Externo  |
+-----+
| + Carregar Dados  |
| + Preprocessar Dados |
| + Treinar Modelo   |
| + Salvar Precisão  |
+-----+
|   Sistema         |
+-----+
| + load_data        |
| + preprocess_data   |
| + train_model       |
| + save_accuracy_to_csv|
```

### *Usuário Externo*

Representa um usuário externo ao sistema, interagindo com suas funcionalidades.

### *Casos de Uso*

#### Carregar Dados

O usuário externo pode carregar os dados do sistema.

#### Preprocessar Dados

O usuário externo pode preprocessar os dados do sistema.

#### Treinar Modelo

O usuário externo pode treinar o modelo do sistema.

#### Salvar Precisão

O usuário externo pode salvar a precisão do sistema em um arquivo CSV.

#### Sistema

Representa o sistema, que contém as funcionalidades implementadas no código.

## Modelo de Dados (MER) para projeto de dados

| Filme | | Gênero |

```

+-----+ +-----+
| - FilmeID: int | | - GeneroID: int |
| - Nome: string | | - Nome: string |
| - Sinopse: string | +-----+
| - Ano: int |
| - Avaliacoes: int |

```

```

| Avaliacao |
+-----+
| - AvaliacaoID: int |
| - FilmeID: int |
| - Estrelas: int |
| - Comentario: string |

```

## Explicações

### *Filme:*

Representa informações sobre um filme, como seu identificador único (FilmeID), nome, sinopse, ano de lançamento e o número de avaliações.

### *Gênero:*

Contém informações sobre os diferentes gêneros de filmes. Cada gênero tem um identificador único (GeneroID) e um nome.

### *Avaliação:*

Registra avaliações feitas pelos usuários para os filmes. Cada avaliação tem um identificador único (AvaliacaoID), o identificador do filme associado (FilmeID), o número de estrelas dado pelos usuários e um possível comentário.

### *Relacionamentos:*

A relação entre Filme e Gênero é representada pela inclusão do GeneroID na tabela Filme, indicando o gênero do filme.

A relação entre Filme e Avaliação é estabelecida pelo FilmeID na tabela Avaliação, associando cada avaliação a um filme específico.