

# You Do #2

İş biriminden gelen talep doğrultusunda adım adım bir recommendation engine oluşturmak istiyoruz. Bunun için de derste kullandığımız MovieLens datasını kullanacağız. Data yı rating matrix formunda yüklemek için aşağıdaki kod bloğunu kullanabilirsiniz

```
df = pd.read_csv('https://files.grouplens.org/datasets/movielens/ml-100k/u.data', delimiter=r'\t',
                 names=['user_id', 'item_id', 'rating', 'timestamp'])

r = df.pivot(index='user_id', columns='item_id', values='rating').values
```

## Model 1

### Part 1

Her bir kullanıcı ve her bir ürün için bir parametre (bias) den oluşan bir model kuruyoruz. Yani;

$$\hat{r}_{ij} = b_i^{user} + b_j^{item}$$

Unutmayalım ki rating matrix üzerinde sadece 100000 değer mevcut. Boş olmayan değerlerin olduğu kümeye  $S$  diyelim

$$S = \{(i, j) : r_{ij} \text{ is observed}\}$$

GD veya SGD kullanarak  $\frac{1}{2} \sum_{(i,j) \in S} (r_{ij} - \hat{r}_{ij})^2$  loss değerini optimize edip optimal parametreleri bulacağız.

### Part 2

Sadece yukarıda verilen form ile gittiğimizde overfit riskine karşı loss fonksiyonunu biraz geliştirelim

$$\frac{1}{2} \sum_{(i,j) \in S} (r_{ij} - \hat{r}_{ij})^2 + \frac{\lambda}{2} \left( \sum_{u=1}^m (b_u^{user})^2 + \sum_{j=1}^n (b_j^{item})^2 \right)$$

Derste de kullandığımız test spliti kullanarak  $\lambda$  hyper parametresini optimize edelim.