**Regression**

A Regression Problem Statement, you would like to compute some continuous values. Ex – Stock returns on a given date, sales on a given week

Quantifying a relationship between 2 or more variables

Ex- Wait time -> Customer Satisfaction

x1 Output

x2 Dependent Variable

x3

Independent

Variables

**Types of Regression:**

* Linear Regression – If mathematical form is linear
* Polynomial Regression – If mathematical form is polynomial
* Non-linear “

**Example: Demand Forecasting**

Need to estimate sales at future point in time.

* A retail business will use it for inventory planning.
* A manufacturing business uses it to plan production cycle.

Sales at a future time might depend on

* Sales in the previous week
* Expected marketing spend – advertisement, promotion
* Holidays – such as Diwali, Eid, Christmas

-Sales in the Sales in

previous week Future week

-Expected Market Spend Dependent variables

-Holidays

Independent variables

**Capital Asset Pricing Model:**

Used for pricing risky securities. Returns of the security to returns of the overall market.

**Ri = Rf + βi(Rm – Rf)**

where Ri = The return on a security

Rf = Risk free rate of return (ex- bonds)

βi(Rm – Rf) = Premium

βi = Volatility of the security ( A measure of how risky the security is)

(Rm – Rf) = Expected return of the overall market over and above the risk free rate.

**Example: Detecting Facial Features:**

Uses:

* Facial Recognition
* Virtual dressing room – seeing sunglass on you eye
* Auto-capture photos – whenever it detects the face

Problem – Find the co-ordinates of the Important facial features

Position depends on(independent variables) –

* Relative position within the picture
* The properties of the surrounding pixels

The co-ordinates of each feature can be found using one regression problem

For example – Left eye centre co-ordinate ( Dependent variables)

**Classification and Regression are similar in many ways:**

Whenever you have a machine learning technique that involves an explicit training then you are performing supervised learning

**Solving Regression Problems:**

**Overview:**

* Example Problem : Understanding how Linear Regression can be applied to find Beta of a stock
* Understand the Stochastic Gradient Method for Linear Regression
* Tweak the parameters of SGD for better performance
* Implement linear regression in python

Objective of regression problem is to quantify the relationship b/w different variables

If you assume the function is linear, then we will use Linear regession

A linear function – b0 + b1x1 + b2x2 + -------

In linear regression the b’s are constant

The whole objective of linear regression is to find the values of these constants. You can solve values of these constants using past data. b’s are called co-efficient.

Dependent variables are the linear combination of independent variables.

Capital Asset pricing model

Ri = Rf + βi (Rm – Rf)

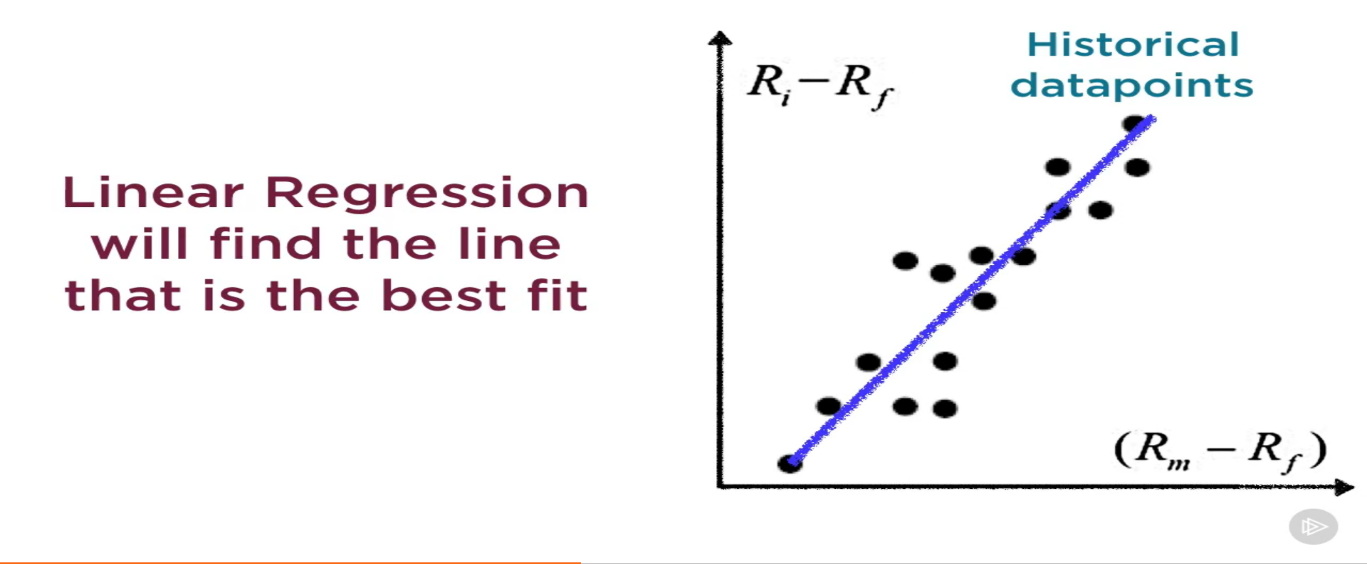
Another form - Ri - Rf = βi (Rm – Rf)

This looks very much like linear regression model

L.H.S-> Dependent Variable (Returns of a given stock)

R.H.S -> Independent Variable ( Overall returns from the market )

Equation of the line passing through origin.



Linear regression helps in finding a line that best fits the given data. Above is a simple linear regression with 1 variable.

Once we find Beta

You can take the market return on that date and multiply with Beta to find the predicted value of the stock’s return.

One Problem – The actual stock return from that day will not be exactly on the line but slightly away from the line. So there will be an error between actual returns and the predicted returns.

Error also called as Residual – Distance between the actual point and the line itself

The goal of any regression problem is to minimize this error over a set of data points called training data. In simple word find a line that will minimize this error b/w the actual value and the predicted value for the training data set.

**Minimizing Error:**

One such technique is Stochastic Gradient Descent. The goal is to minimize error.

We are taking mean squared Error:

Error = 1/N \* ∑i=1N €2

N -> number of historical datapoints

Go to Andrew Ng Notes

**Example: The CAPM model for Google Stock**

Lets say you wanted to find Beta for Google

Note: Google returns is the dependent variable and Nasdaq return is ind. var.

Ri - Rf = βi (Rm – Rf)

Actual value of βi = 1.03

We will need

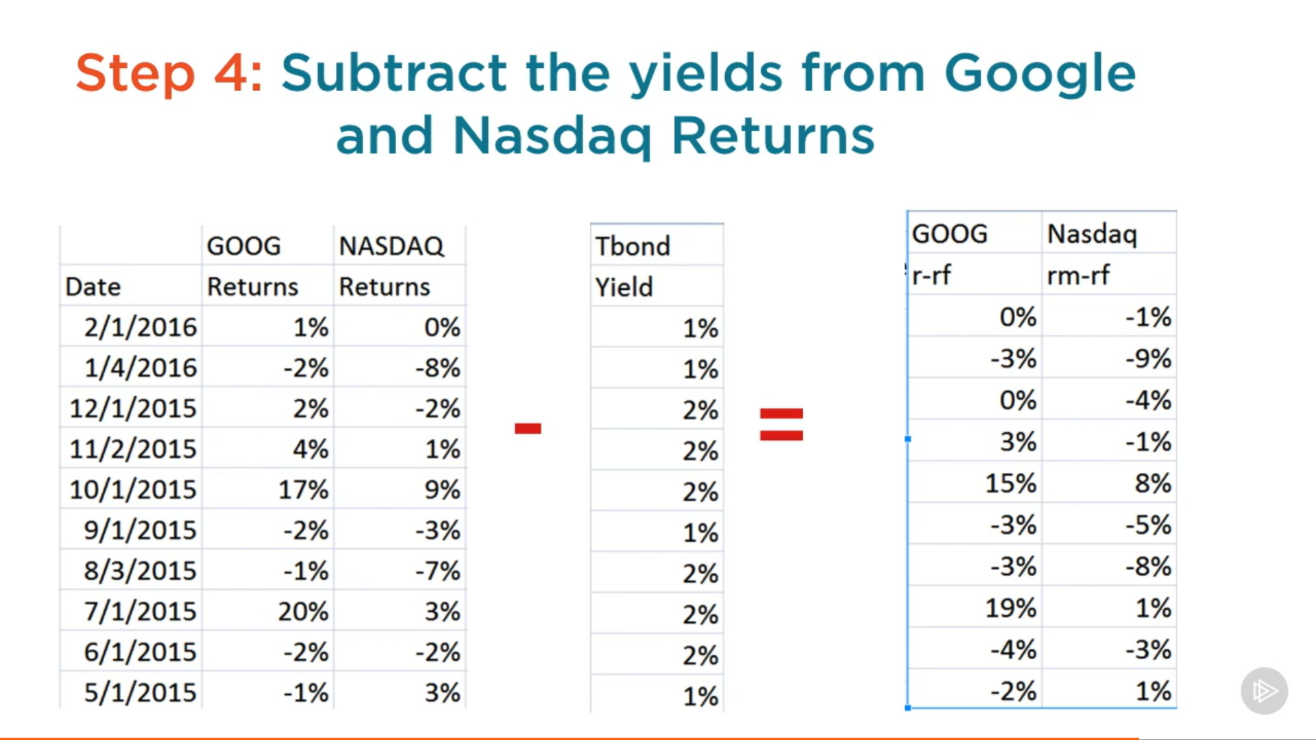
* Ri - Rf (Returns of Google )
* (Rm – Rf) ( Returns of an Index that represents the market – Risk free Rate)We will use NASDAQ index

**Steps:**

* Download Historical Prices for Google and Nasdaq from a financial site (Yahoo Finance)
* Converts the prices to returns (Returns is what would you get if you invested in the beginning of the month and then sold it at the end of month)

Monthly Return = (New Price – Old Price ) / Old Price

* Compute the risk free rate of return using the yields of 5 year Treasury bonds
* Subtract the yields from Google and Nasdaq Returns



* Regress the adjusted Google returns against adjusted Nasdaq returns

**Recommending Relevant Product to a User:**

**Overview:**

* Understand the role of Personalized Recommendations
* Predict user-product ratings using Collaborative Filtering
* Find hidden factors that influence user-product ratings
* Implement Movie Recommendations in Python

Example:

* Product recommendations on Amazon
* Movie recommendation on netflix
* Gmail auto tagging important e-mail

The new trend across all online services (Personalization)

* Inbox organization
* Facebook newsfeed
* New York Times homepage

Current problem, Offline stores are designed to appeal to the majority of users

What if, the store could change, so that each user sees the design that most appeals to them? (This is not possible in real life but it is possible in online service)

The store is personalized based on specific user

* The user’s preferences
* The user’s needs
* What he currently looking or searching for

All the online services like Amazon, Netflix, Youtube are 100 of thousands of videos.

**Discovery:**

So, User need help in finding what they are looking for, …sometimes things they didn’t know they were looking for

**Engagement:**

The more time users spend at a website, the more likely they are open their wallet to buy something or do something.

**Recommending Relevant Product to a user: Predicting rating using Collaborative Filtering:**

**Example 1: Find top 10 Movie Picks for a User**

**Steps:**

* Compute the rating for all movies for all Users
* Sort movies for each user in descending order based on their ratings
* Pick the top 10 movies this user has not watched yet

**Example 2: Recommendation based on Browsing History**

**Steps:**

* Use # views of a Product as an implicit rating
* Compute the ratings for all products for all users
* Pick the top 10 products for this user

**Example 3: User Who Bought This also Liked…**

**Steps:**

* Compute the ratings for all products for all users
* Subset the computed ratings to users who bought this product
* Pick the 10 products with highest ratings for this subset

**The Common Problem to Solve:**

**Compute rating of all products for all users**

**Collaborative Filtering Algorithms:**

Predict user rating for products based on a user’s past behaviour

-User purchases -Top picks for you

Collaborative

Filtering

-User Browsing History -If you like this,

-User Clicks you’ll love that!!

-User Ratings, Reviews -If you have bought

this, you’ll need that!

A general term for any algorithm that only uses past user behaviour for identifying recommendations.

The basic premise is that:

* If 2 users have the same option about a bunch of products, They are likely to have the same opinion about other products too!

The training data should be in the form of

User Id

Product Id

Rating

Product could be – Books, videos, Movies, Artists, News Articles, e-mail etc

Rating – A measure of user’s preference for a particular product

Types:

* Explicit Rating – Collected at the store or through e-mail surveys
* # Clicks, # Purchases, # Shares, # Likes, # Times watched

**Recommending Relevant Product to a user: Finding Hidden Factors that influence Ratings**

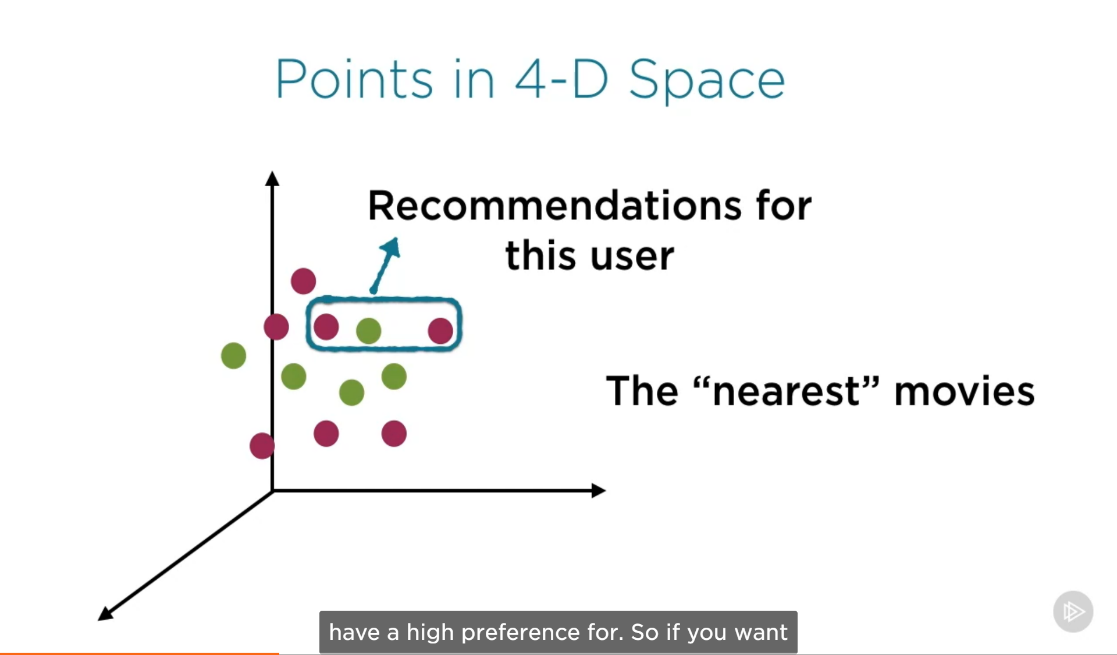
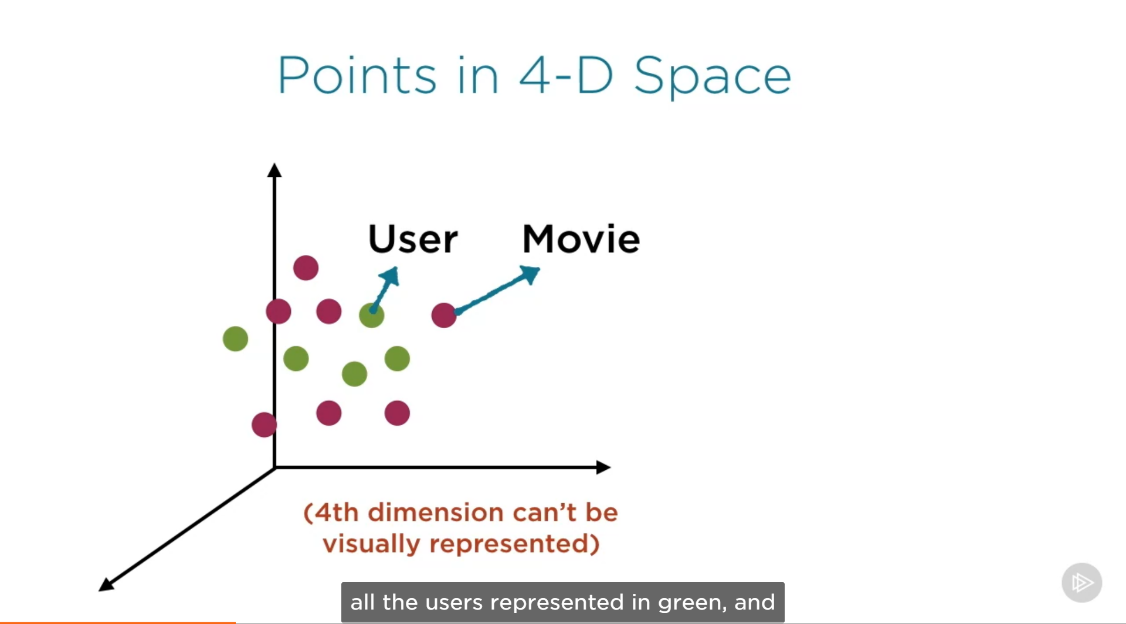
**Hidden factors for a movie:**

Genre, Cast Popularity, Commercial Appeal, Recency of Release

1. Rate every User on their preferences for these above factors
2. Rate every movie on a scale of 1 to 5 for these above factors

So every user and every movie is represented using 4 numbers

Points in 4D space



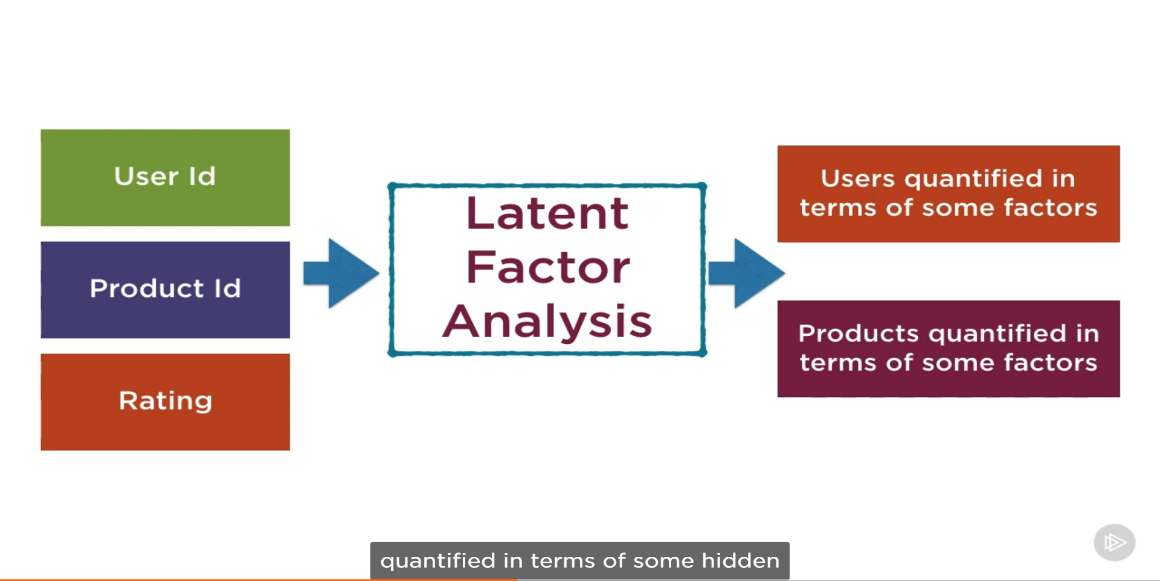
The key step in this approach – Representing users and movies in terms of influencing factors.

**Assumptions:**

* We know which factors influence user’s preferences
* We can quantify factors for each user
* We can quantify those factors for each movie

These data are actually difficult to get in real life.

**So we will take help of Latent Factor Analysis Technique**

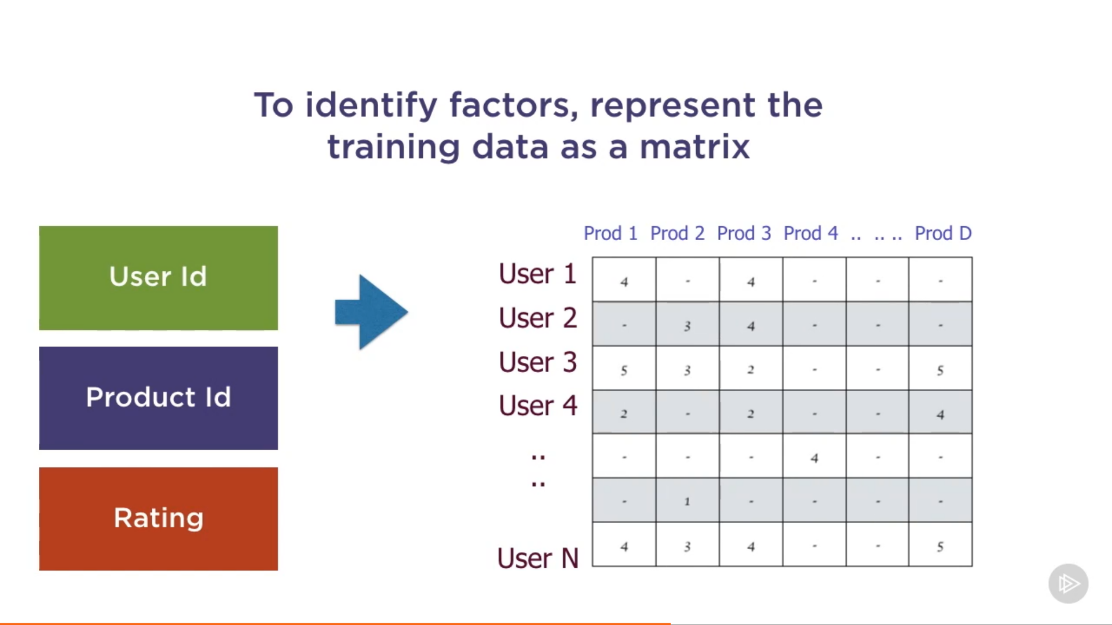
****

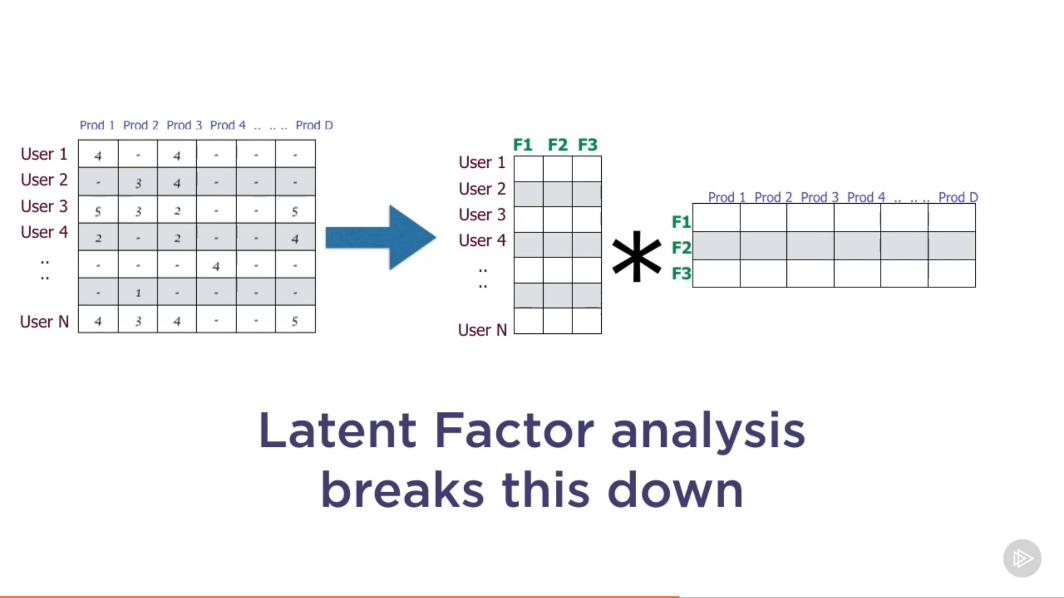
**Why it is called Latent Factor Analysis ?**

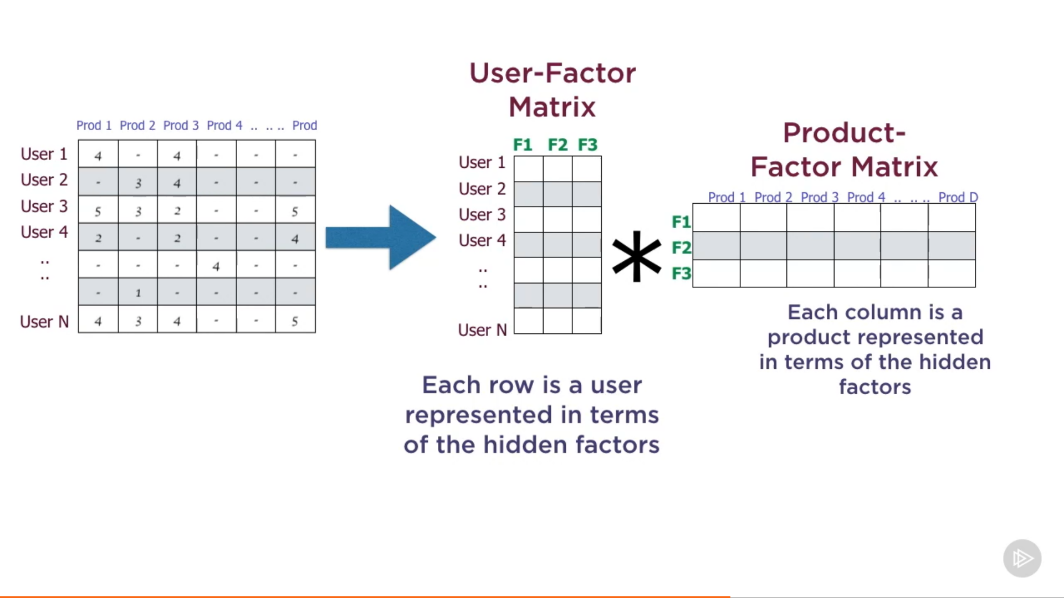
The factors are not known beforehand. We are not hypothesising that there is a genre or task popularity factor that actually influences user’s rating but letting the user’s rating data tell us what the factors actually are

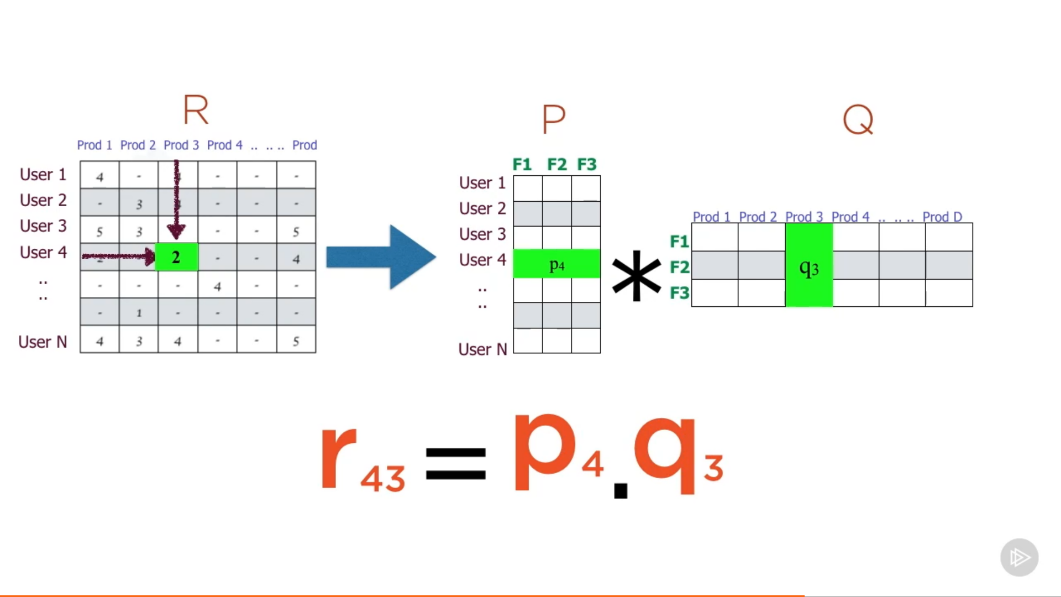
Once they are identified, they might turn out to be

* Factors with meaning like genre, cast popularity
* Abstract factors with no real life meaning

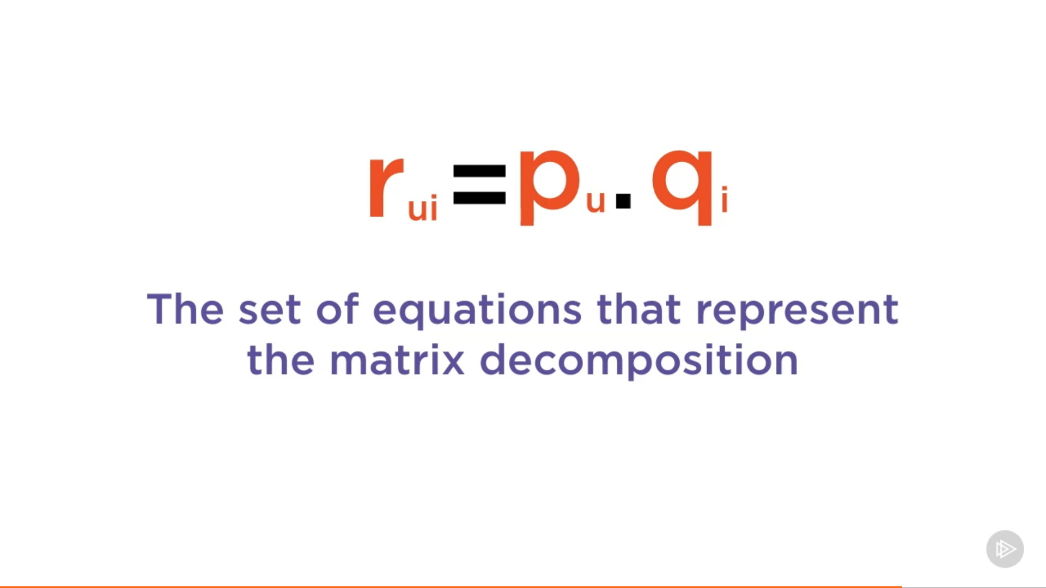


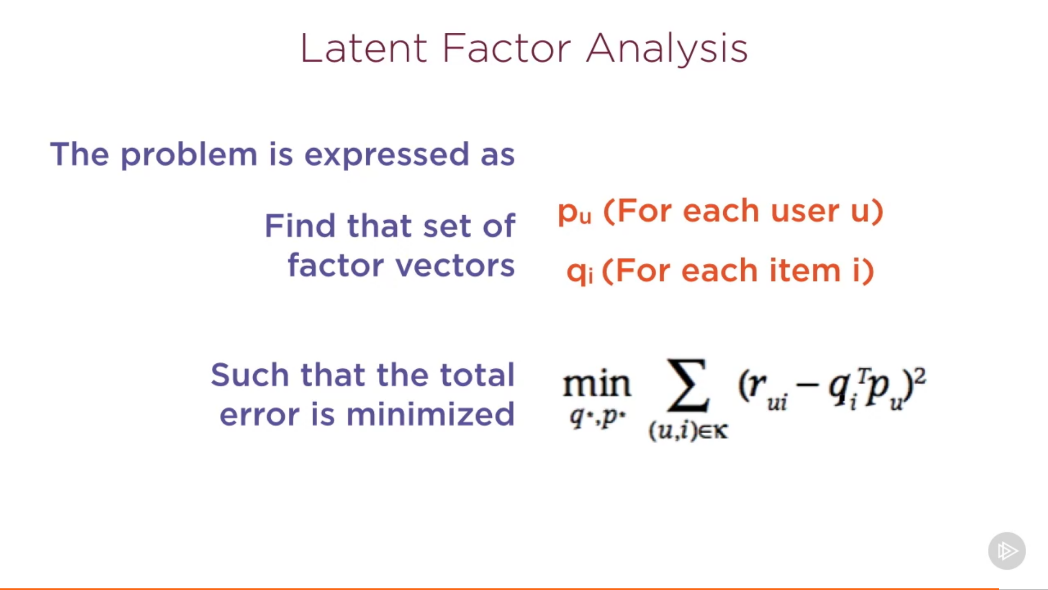






If you see the first picture you can see that all users have not made ratings for all the products many rating cells are empty. So we take the rating which we actually have decompose that matrix and from the decomposed matrices we can compute what the ratings could be for any product.





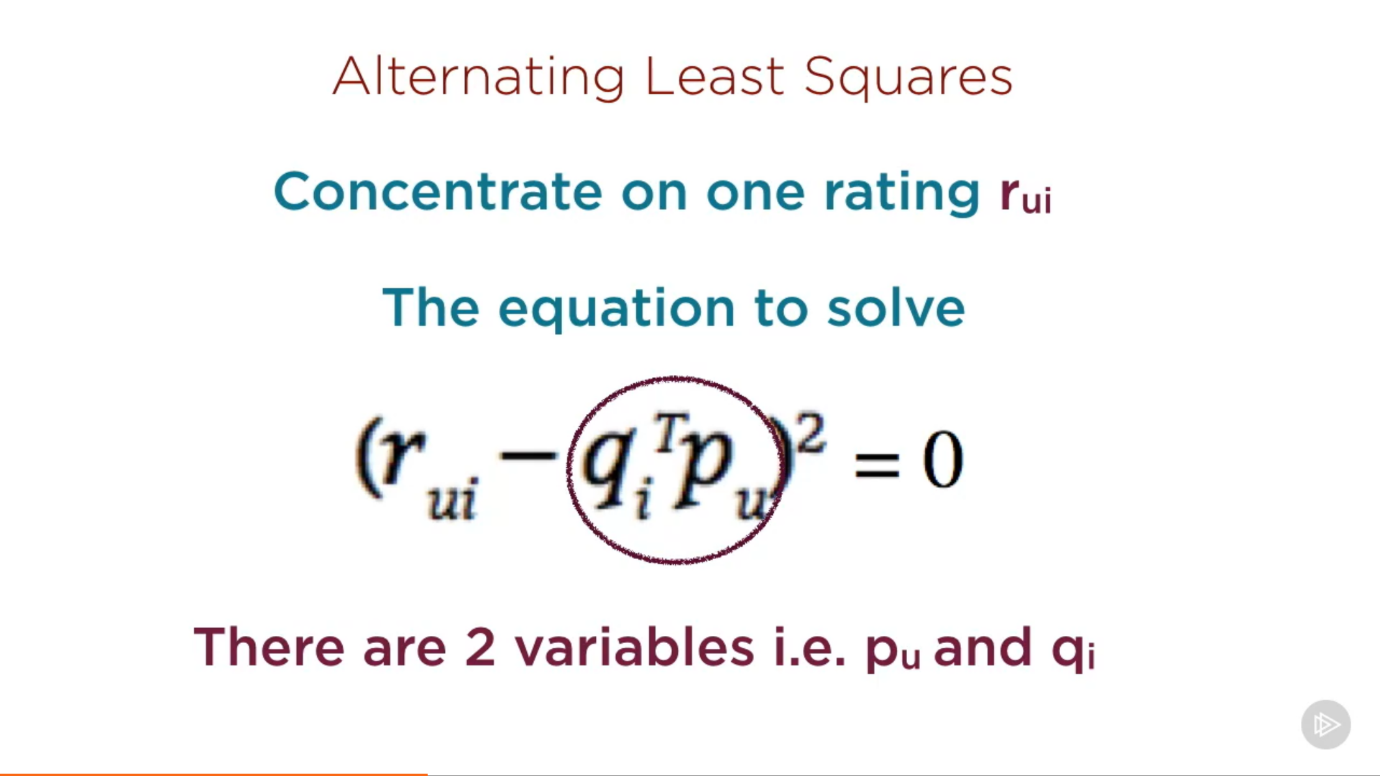
Alternatively Least Squares is a technique to minimize this error

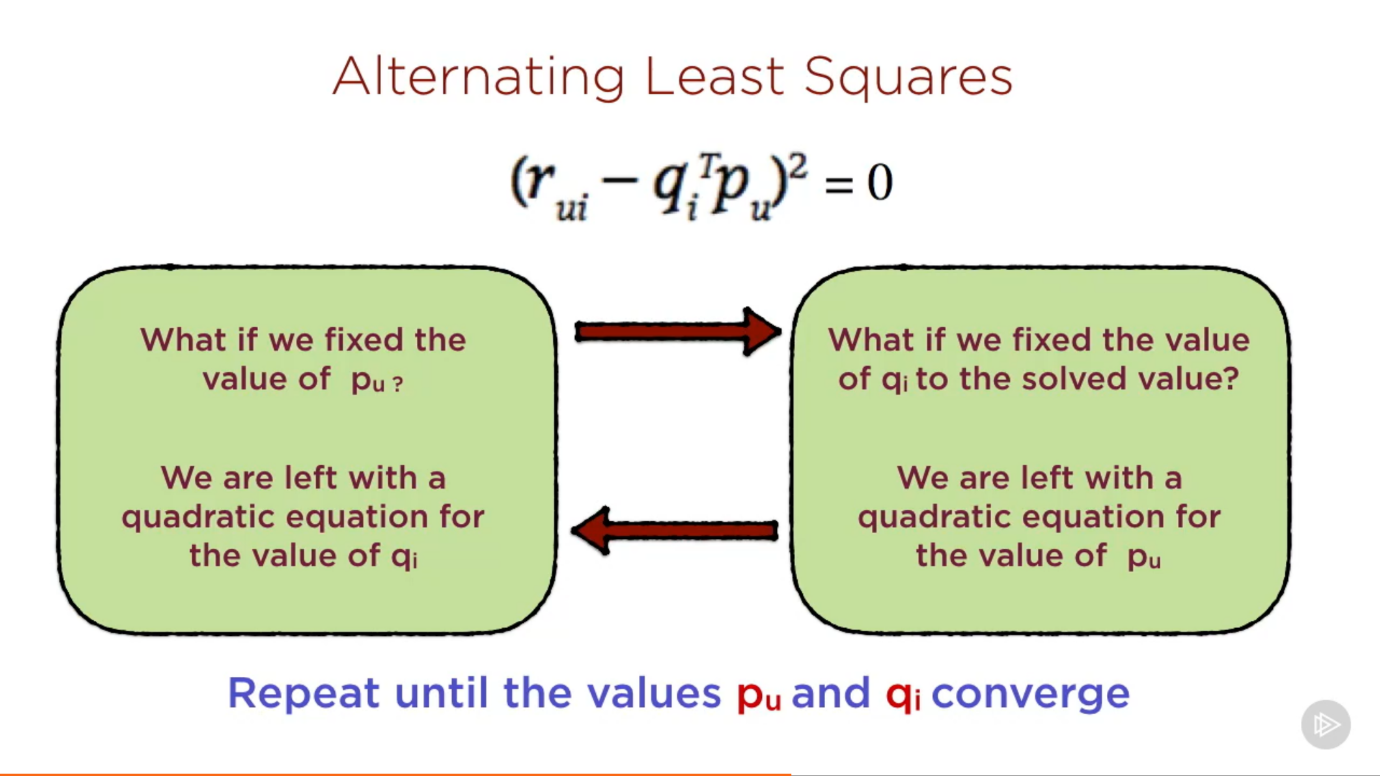
min ∑(u, i )€k ( rui – qiT pu )2

rui = actual rating for the user for a particular item

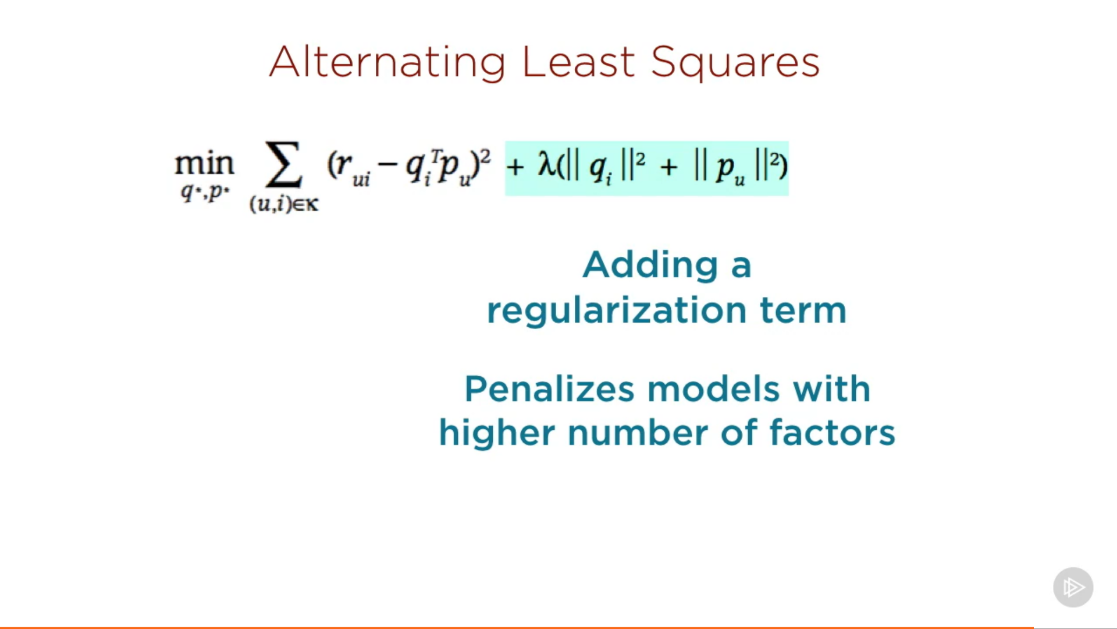
qi = item factor vector

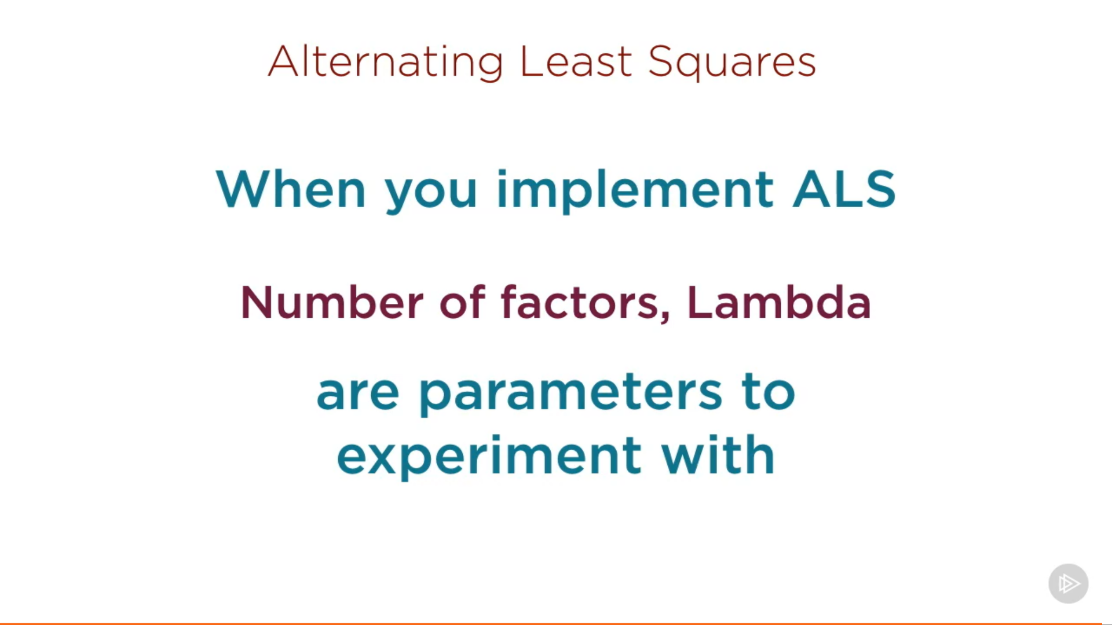
pu = product factor vector





The choice of the number of hidden factors is left to the user. (The more the no of factors you use the more is the complexity of model). In general in machine learning, we want to find simpler model. If a simpler model can explain the same relationship, then we would like to choose that.





**Implementing ALS to Find Movie Recommendations:**