

Deep Learning for Recommender Systems

Strata Data Conference
New York 2017

Recommender Systems
Collaborative Filtering
Deep & Wide Learning
Deep Matrix Factorization

Mo Patel
Director, Artificial Intelligence
Think Big Analytics
Cynthia Li
Data Scientist
Think Big Analytics

bit.ly/strata17recsys

Agenda

- Recommender Systems 101

- Collaborative Filtering

- Deep & Wide Learning

- Deep Matrix Factorization

History of Recommendation



Human

- DJ
- Librarian
- Sommelier

Human + Data

- Market Basket
- People you may know
- Job you may be interested in
- Items you may want to consider

Data + Machines

- Deep & Wide Learning
- Deep Matrix Factorization

“

A family of methods that enable filtering through large observation and information space in order to provide recommendations in the information space [whether] that user does [or does] not have any observation, where the information space is all of the available items that the user could choose or select and observation space is what user experienced or observed so far.

”

Bugra Akyildiz on Recommender Systems

Recommender Systems Impact



2/3 of the movies watched are recommended



Recommendations generate 38% more click through on Google News

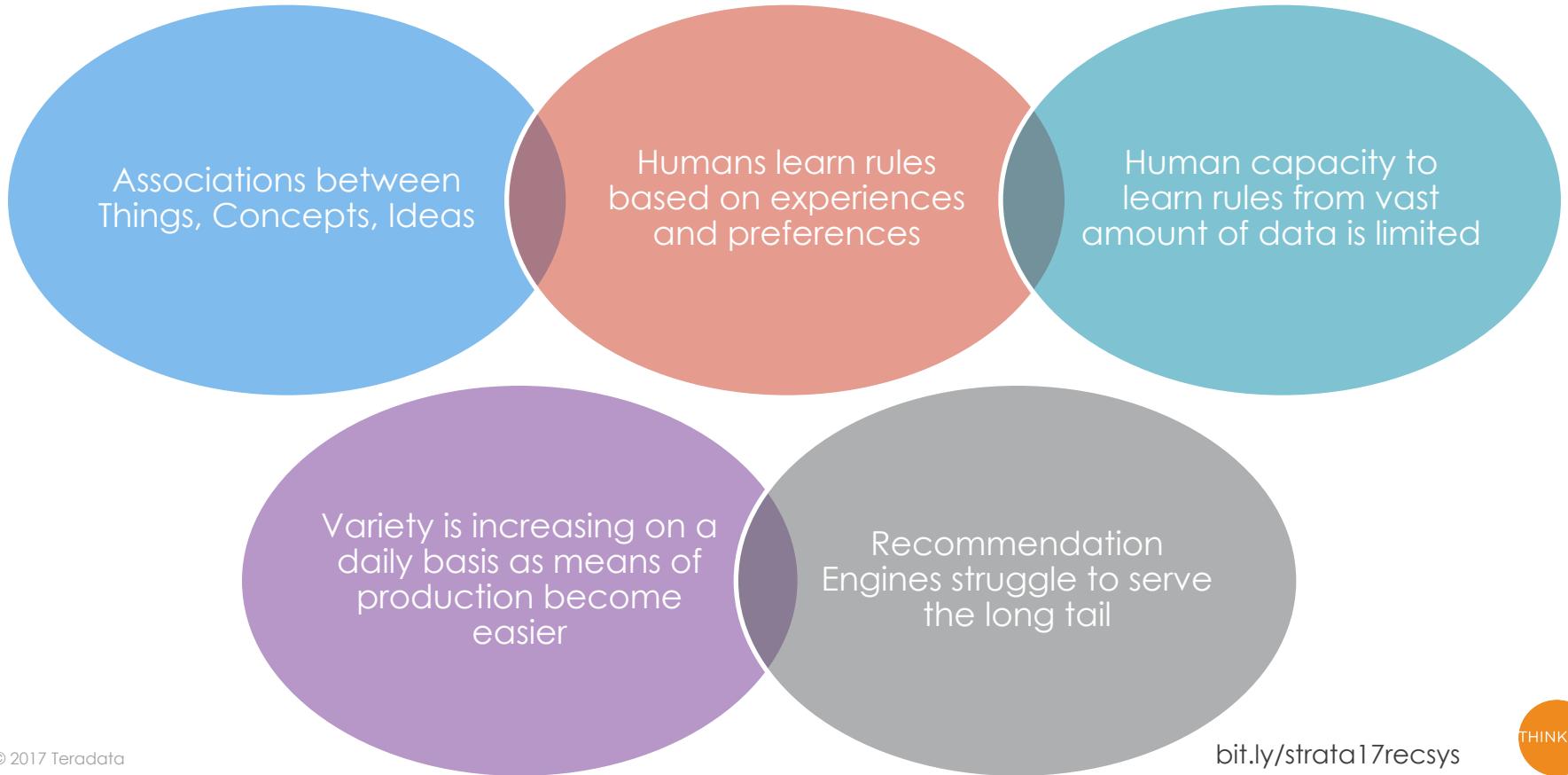


70% of Amazon home page is devoted to Recommendations



"It's scary how well Spotify Discover Weekly playlists know me."
@Dave_Horwitz

What are recommender systems?



Recommendation System Challenge

Before

- Few Users, Few Items

Users	Item1	Item2	Item3	Item4	Item5
User1	x		x		x
User2	x	x	x		
User3			x	x	x
User4	x	x			x

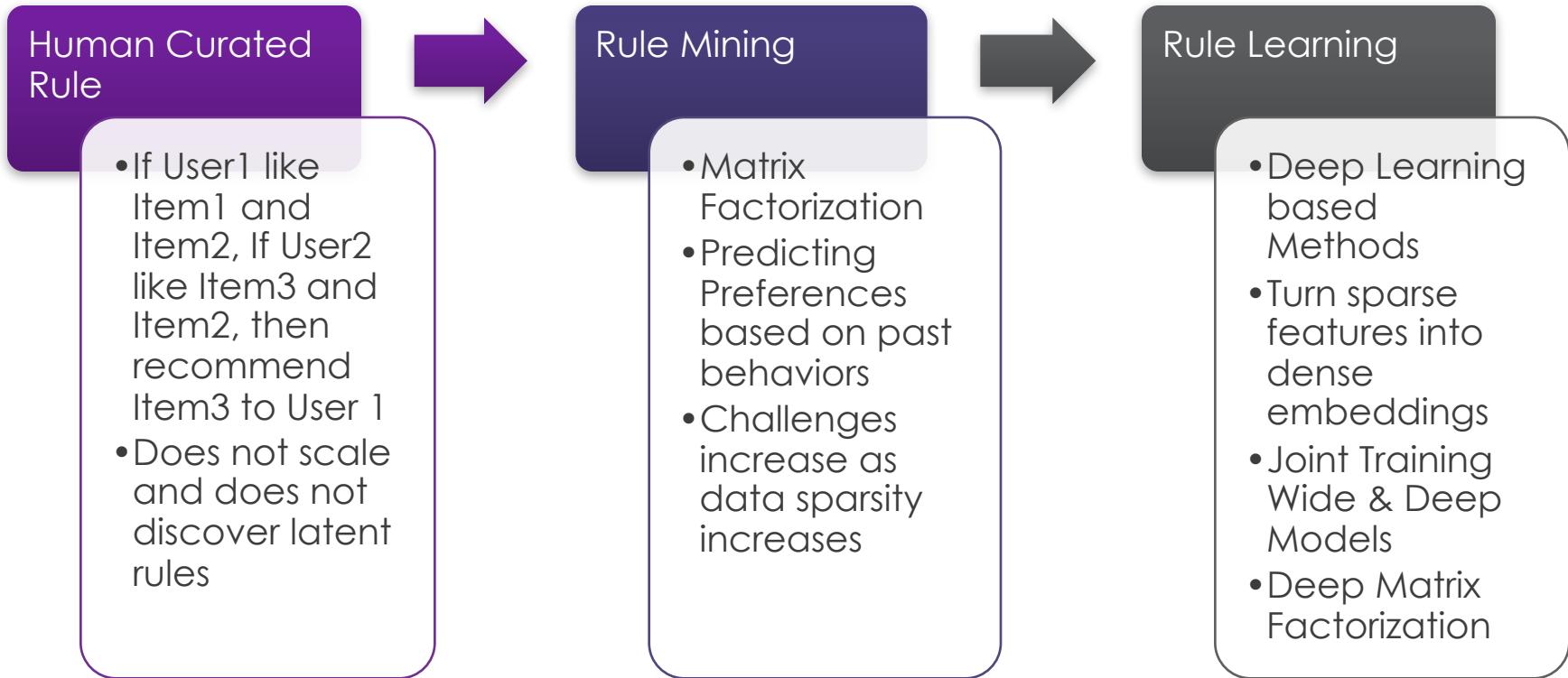
Today

- Millions & Billions of Users & Items

Users	Item1	Item2	Item3	...	m
User1	x		x		x
User2	x	x	x		
User3			x	x	x
...				x	x
n	x	x			x



Recommendation System Rules



Agenda

- Recommender Systems 101

- Collaborative Filtering

- Deep & Wide Learning

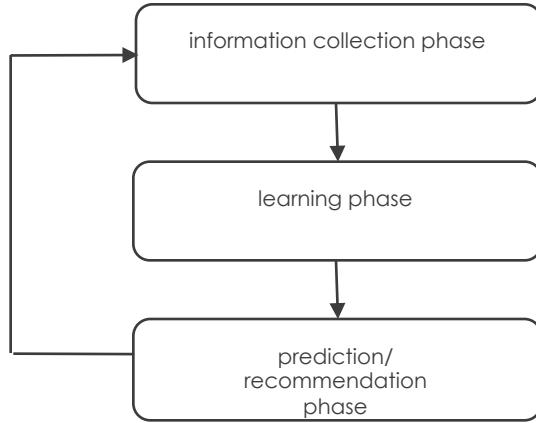
- Deep Matrix Factorization

Traditional recommendation approaches

- Collect data -> explicit/implicit/hybrid feedback
- Learn Rules
 - Wine & Cheese
 - Beer & Diapers
- Collaborative Filtering

Traditional recommendation approaches

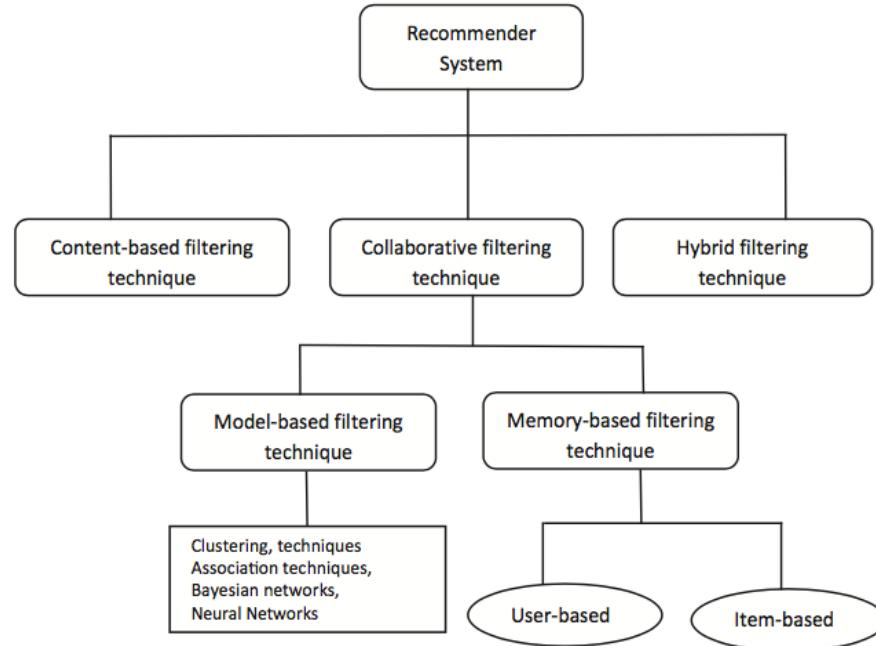
- Collect data -> explicit/implicit/hybrid feedback



Recommendation phases

Traditional recommendation approaches

- Learning phase -> prediction recommendation



source: Recommendation systems: Principles, methods and evaluation

Collaborative Filtering

Intuitively:

- Personal tastes correlate for a given domain and information space.
- Users who agreed before are more likely to agree in future on some particular item.
- To approximate the rating of the user, we select users who have somehow similar taste.

Collaborative Filtering

Advantages over content based methods

- No need to know about item content
 - Content of the item does not necessarily tell the whole story. What features(length, category, actor/actresses) of movie of an item is important for a particular user? No information is necessary for CF.
- "Item cold-start" problem is avoided
 - When the item information is not available, we do not need it as long as some users purchased the item.

Collaborative Filtering

Advantages over content based methods

- No need to know about item content
 - Content of the item does not necessarily tell the whole story. What features(length, category, actor/actresses) of movie of an item is important for a particular user? No information is necessary for CF.
- "Item cold-start" problem is avoided
 - When the item information is not available, we do not need it as long as some users purchased the item.

Cold start is a potential problem in computer-based information systems which involve a degree of automated data modelling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information.

Collaborative Filtering

Advantages over content based methods

- User interest may change over time.
 - Focusing on content solely does not provide any flexibility on the user side. We need to take the user preference change into consideration as well.
- Explainability
 - Rather than rely on content, plausible social rationale for the user can result (eg: "You will like this product because other users like you like the product").
- Capture subtlety
 - This is generally true for latent factor models. If most of the users are buying two unrelated items, then it is likely another user who shares the same taste of the larger set of users is likely to buy that unrelated thing, which brings serendipity to the table.

Collaborative Filtering

Disadvantages vs content based methods

- Sparsity of user preference and Scalability in computation
 - User preference sparsity makes it hard to find users who share similar taste
 - Large sparse matrices make hard to do computation in general.
- Privacy Concerns
 - Some users may not want their preference and taste available on the website even in an implicit form.
- Synonym issue needs to be handled properly and normalized
 - Film-movie corresponds to the same thing, if one user buys comedy and the other buys "funny movies", they should be treated as same in order to improve both recommendations and not separate the things that are inherently alike.

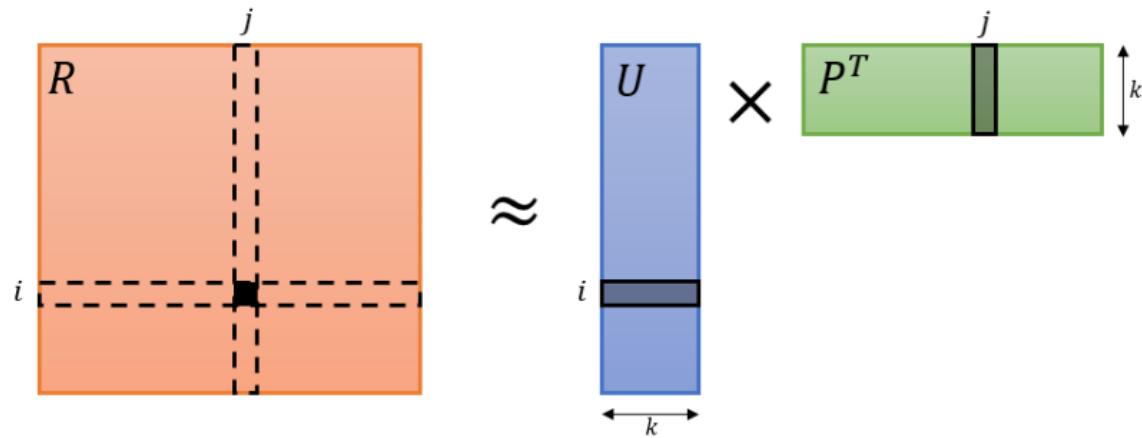
Collaborative Filtering

Disadvantages vs content based methods

- Gray Sheep
 - There may be people whose taste and preference do not consistently agree with other users' preference. Recommendations for these users may not make sense as their taste is quite distinct from others.
- Shilling Attack
 - People who like one brand may give consistently high scores to that brand and give poor scores to the competitors no matter what their real experience are.

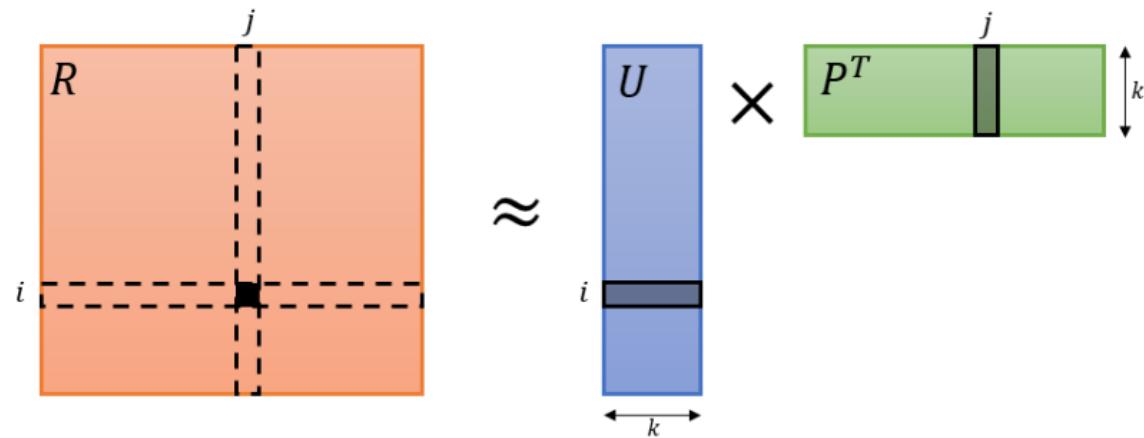
ALS Method for Collaborative Filtering

- Matrix Factorization



ALS Method for Collaborative Filtering

- Matrix Factorization



$$J = \|R - U \times P^T\|_2 + \lambda (\|U\|_2 + \|P\|_2)$$

<https://datasciencemakesimpler.wordpress.com/tag/alternating-least-squares/>

ALS Method for Collaborative Filtering

- Alternating Least Squares

$$\|R - U \times P^T\|_2 = \sum_{i,j} (R_{i,j} - u_i \times p_j)$$

Solve for β by minimizing square errors: $\|y - X\beta\|_2$

Matrix Factorization Notebook

- Demonstrate how to use the TensorFlow API to implement a matrix factorization model (ALS)

bit.ly/strata17recsys

Our Dataset

- Our dataset: MovieLens 1m dataset
 - <http://files.grouplens.org/datasets/movielens/ml-1m.zip>



The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** "jupyter" logo and "Logout" button.
- Navigation:** "Files", "Running", and "Clusters" tabs.
- Toolbar:** "Upload", "New", and a refresh icon.
- File List:** A table showing the contents of the "ml-1m" directory. The table has columns for selection checkboxes, file/folder icons, file names, and last modified times.

		Name ↑	Last Modified ↑
<input type="checkbox"/>	seconds ago
<input type="checkbox"/>	movies.dat		14 years ago
<input type="checkbox"/>	ratings.dat		15 years ago
<input type="checkbox"/>	README		2 years ago
<input type="checkbox"/>	users.dat		15 years ago

bit.ly/strata17recsys

Our Dataset

- movies.dat

1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance
4::Waiting to Exhale (1995)::Comedy|Drama
5::Father of the Bride Part II (1995)::Comedy

- users.dat

1::F::1::10::48067
2::M::56::16::70072
3::M::25::15::55117
4::M::45::7::02460

- ratings.dat

Columns are separated by ::

1::1193::5::978300760
1::661::3::978302109
1::914::3::978301968
1::3408::4::978300275



- Gender is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:

- * 1: "Under 18"
- * 18: "18-24"
- * 25: "25-34"
- * 35: "35-44"
- * 45: "45-49"
- * 50: "50-55"
- * 56: "56+"

- Occupation is chosen from the following choices:

- * 0: "other" or not specified
- * 1: "academic/educator"
- * 2: "artist"
- * 3: "clerical/admin"
- * 4: "college/grad student"
- * 5: "customer service"
- * 6: "doctor/health care"
- * 7: "executive/managerial"
- * 8: "farmer"
- * 9: "homemaker"
- * 10: "K-12 student"
- * 11: "lawyer"
- * 12: "programmer"
- * 13: "retired"
- * 14: "sales/marketing"
- * 15: "scientist"
- * 16: "self-employed"
- * 17: "technician/engineer"
- * 18: "tradesman/craftsman"
- * 19: "unemployed"
- * 20: "writer"

The Goal

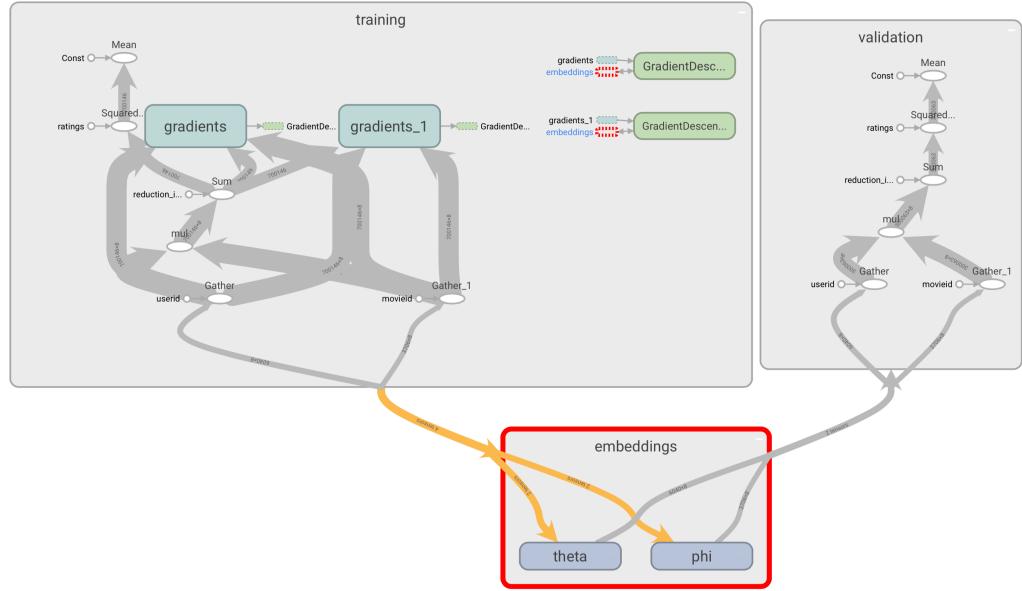
- Our Goal is to predict the rating for a given pair of user id and movie id

gender	userid	movieid	age_desc	occ_desc	title	genre	rating	prediction (rnd.)	prediction (prc.)
49536	M	2507	171	25-34	college/grad student	Jeffrey (1995)	Comedy	4	4
299582	M	984	1197	50-55	self-employed	Princess Bride, The (1987)	Action Adventure Comedy Romance	4	4
738324	F	3308	2724	18-24	writer	Runaway Bride (1999)	Comedy Romance	4	3
900461	M	5795	3432	25-34	academic/educator	Death Wish 3 (1985)	Action Drama	1	1
531555	M	5627	1967	25-34	other or not specified	Labyrinth (1986)	Adventure Children's Fantasy	2	3

MF Accuracy: 45.008881%

Implementation Steps

- Load data
- Set up our parameters: θ and φ
- Define non-trainable variables: user ids, movie ids and ratings
- Calculate the prediction and cost
- Train and evaluate the model



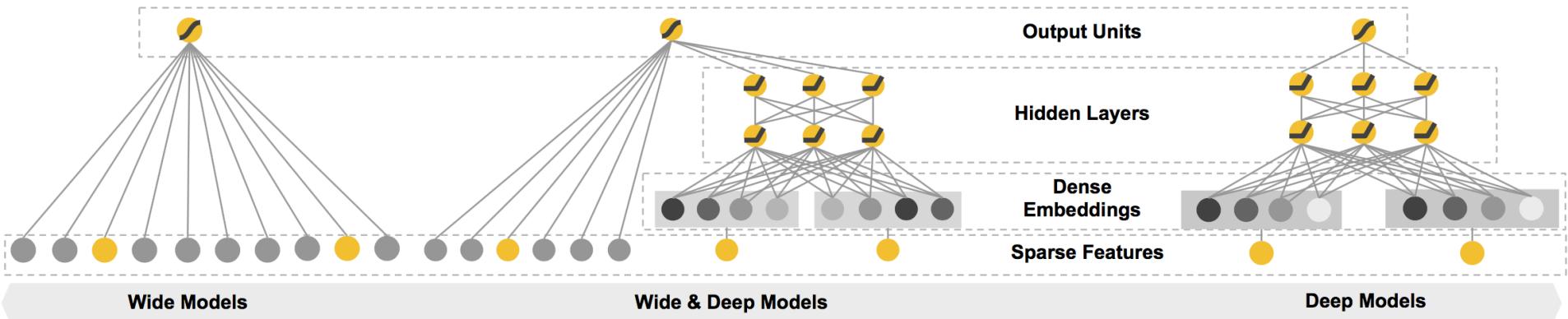
tensorboard --logdir=path/to/log-directory

Agenda

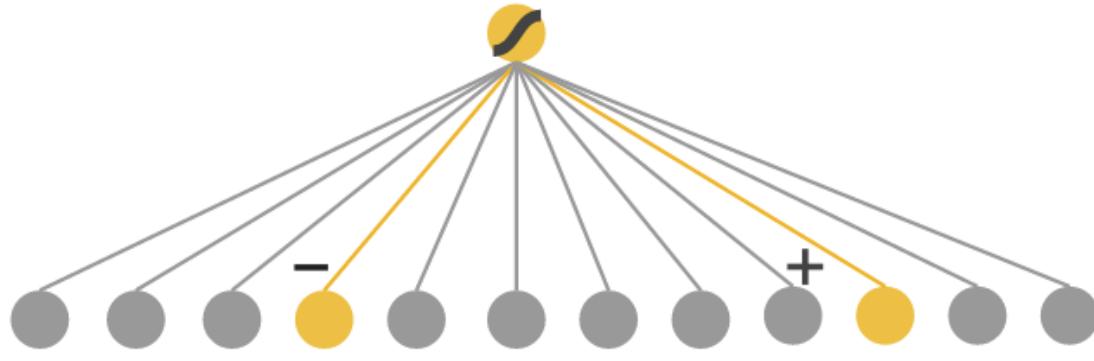
- Recommender Systems 101
- Collaborative Filtering
- Deep & Wide Learning

- Deep Matrix Factorization

Wide & Deep Learning for Recommender Systems



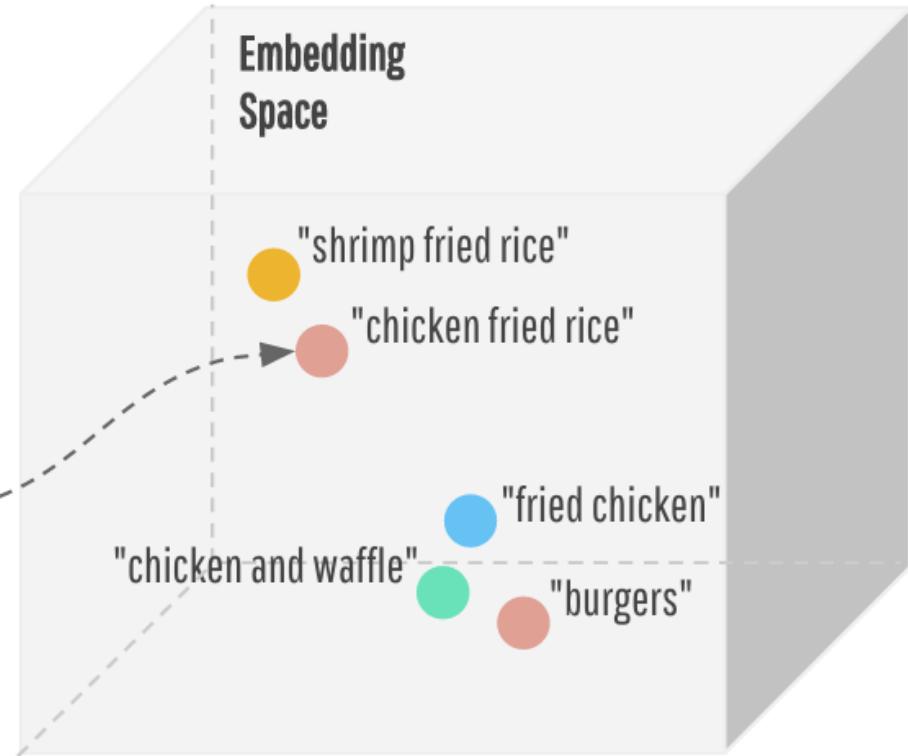
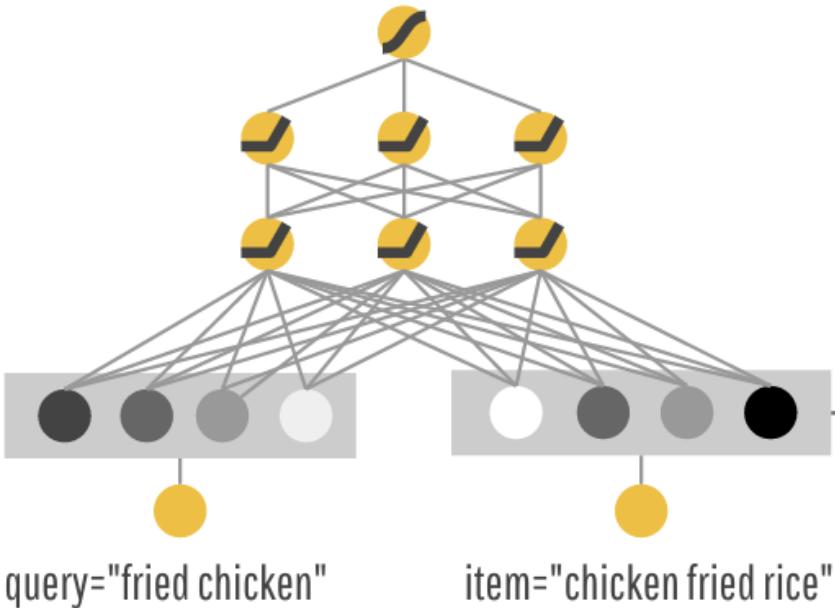
Wide Model



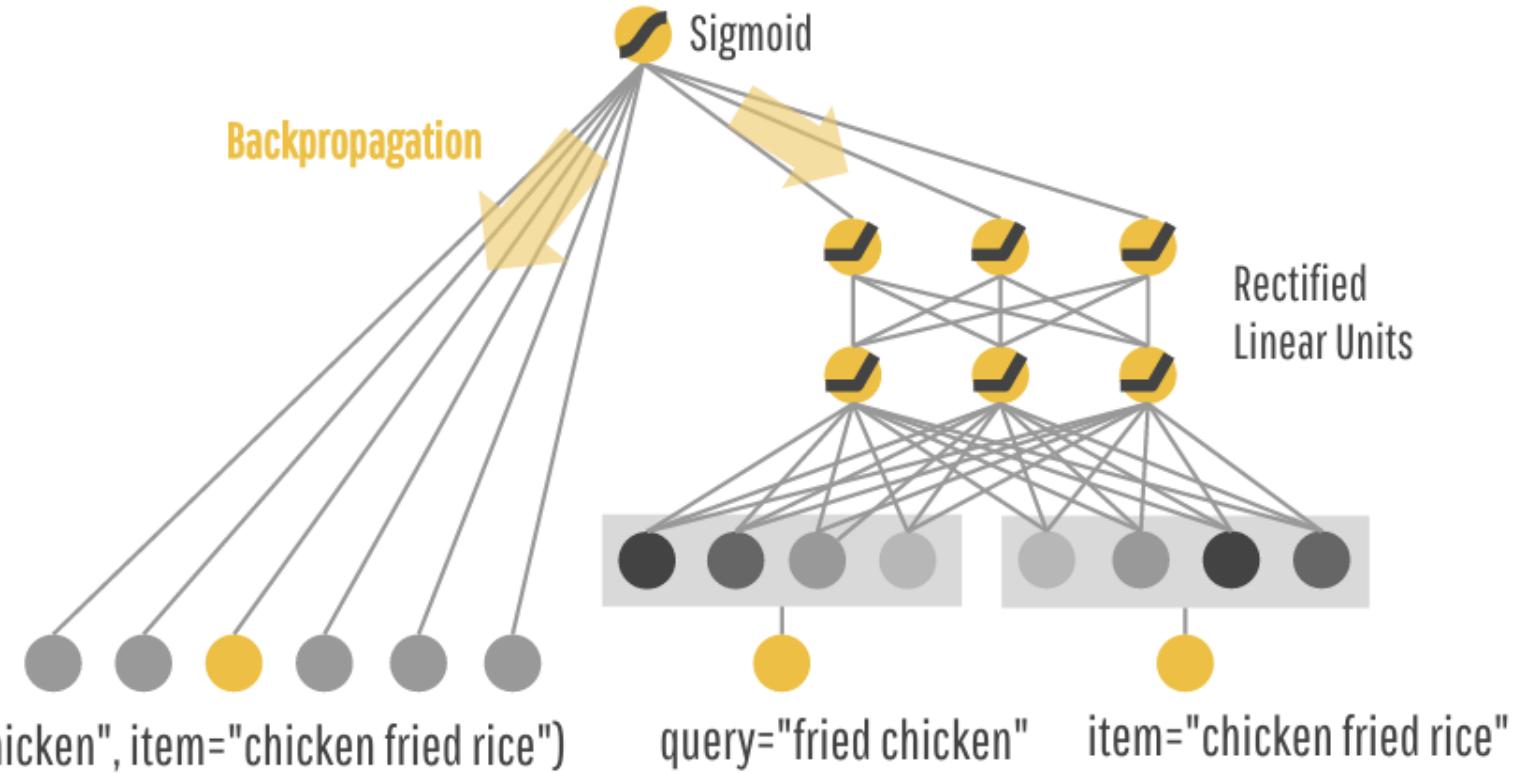
AND(query="fried chicken", item="chicken fried rice")

AND(query="fried chicken", item="chicken and waffle")

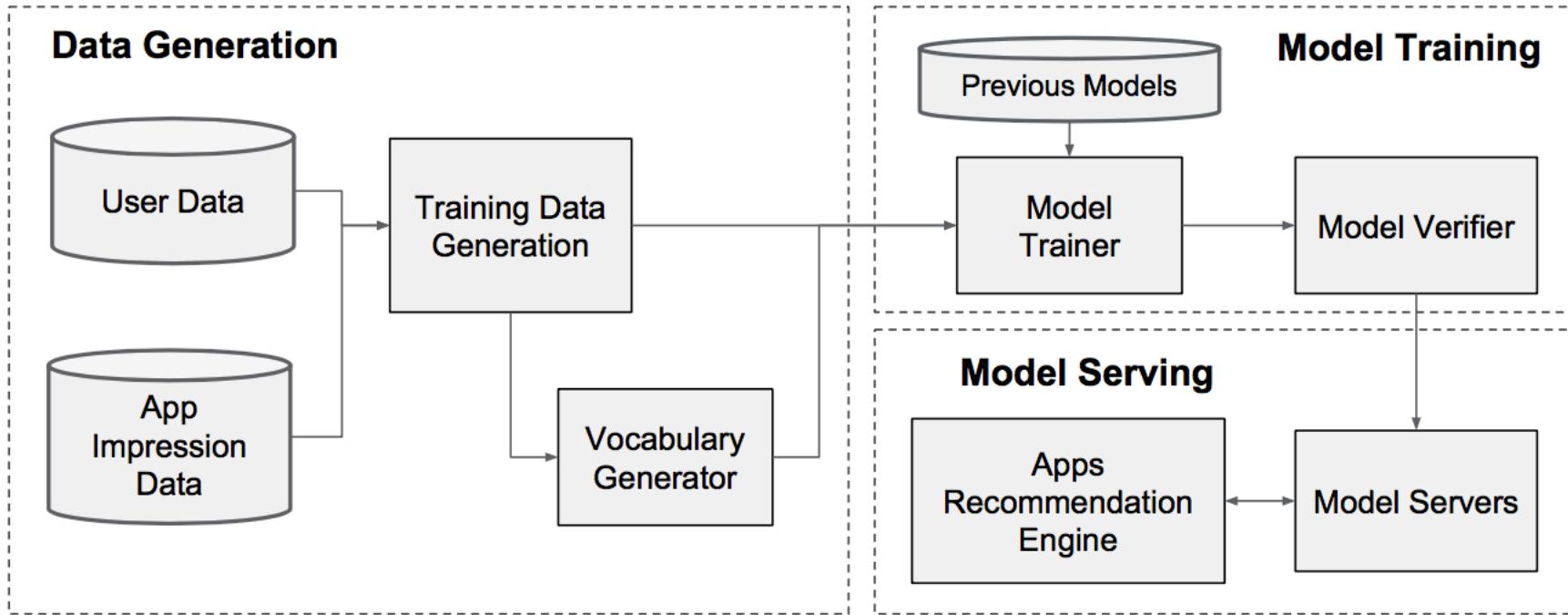
Deep Model



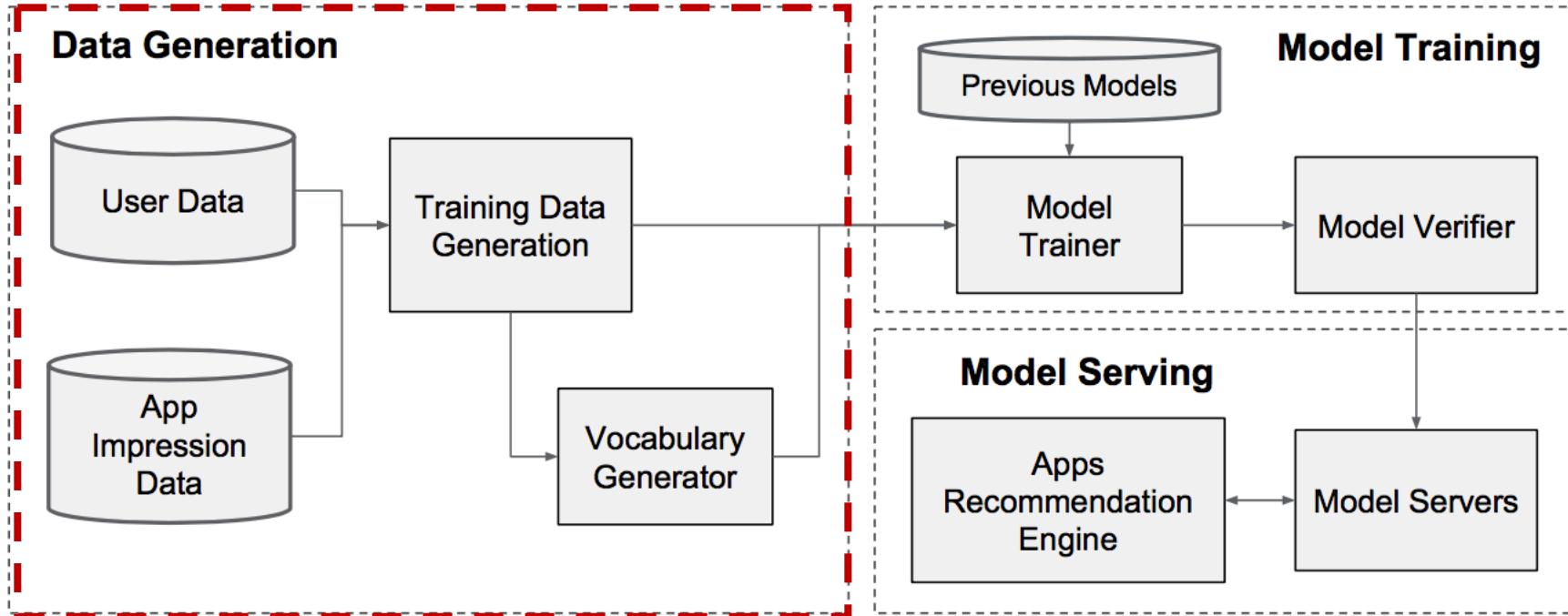
Wide & Deep Model

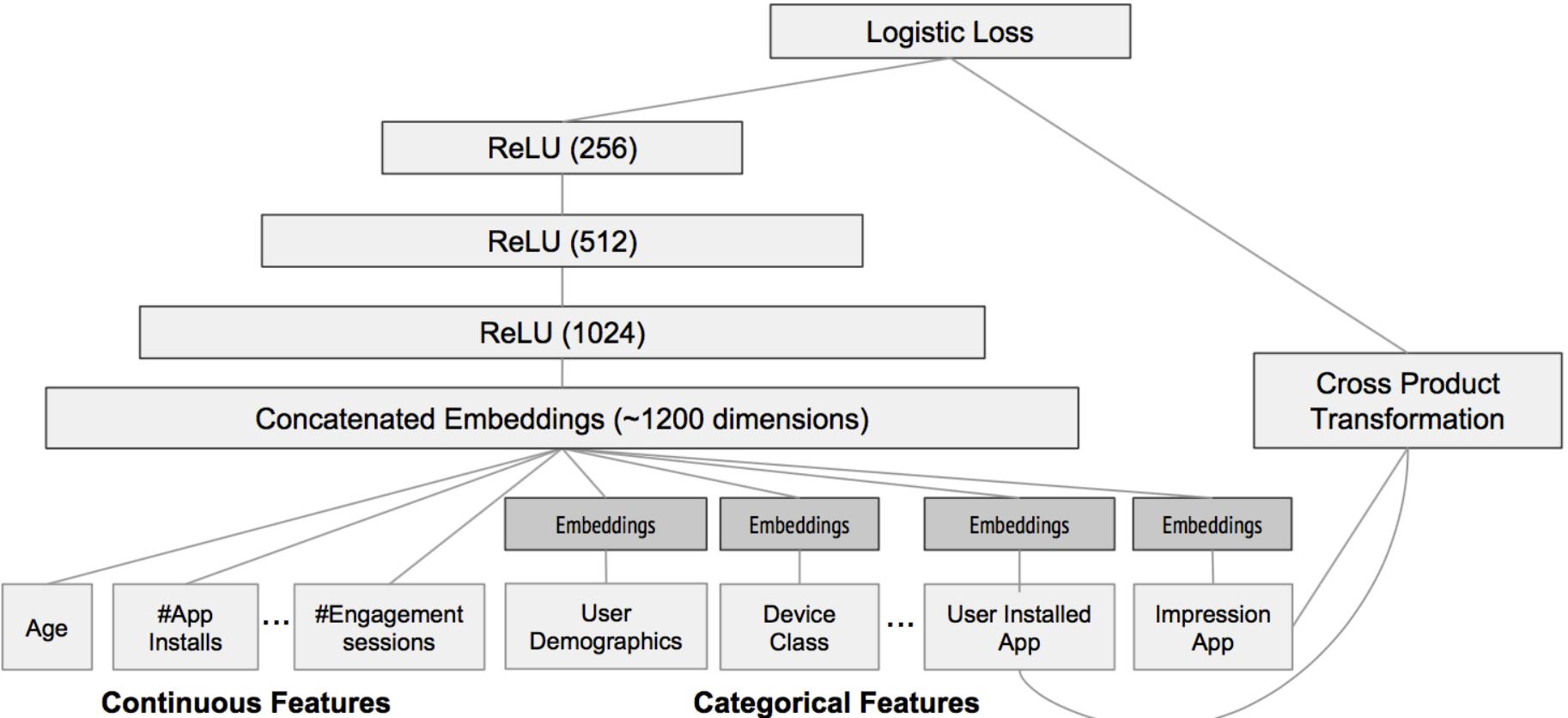


Wide & Deep

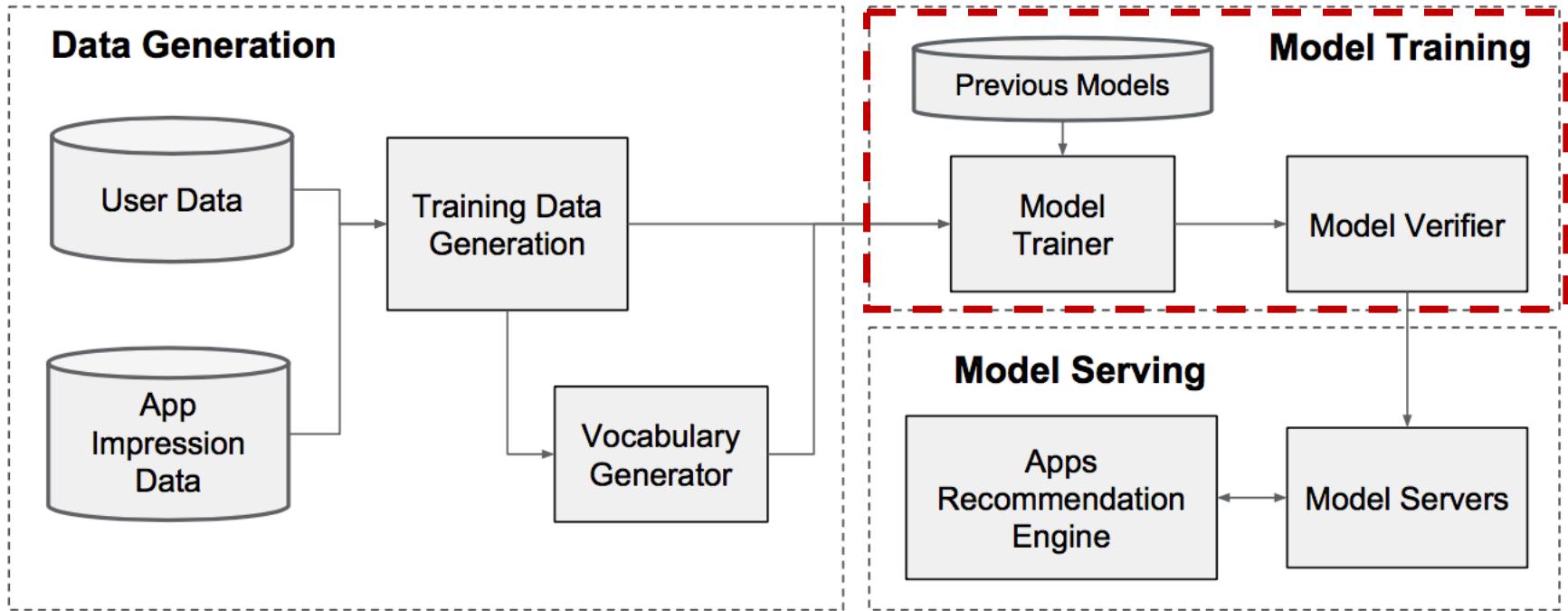


Wide & Deep

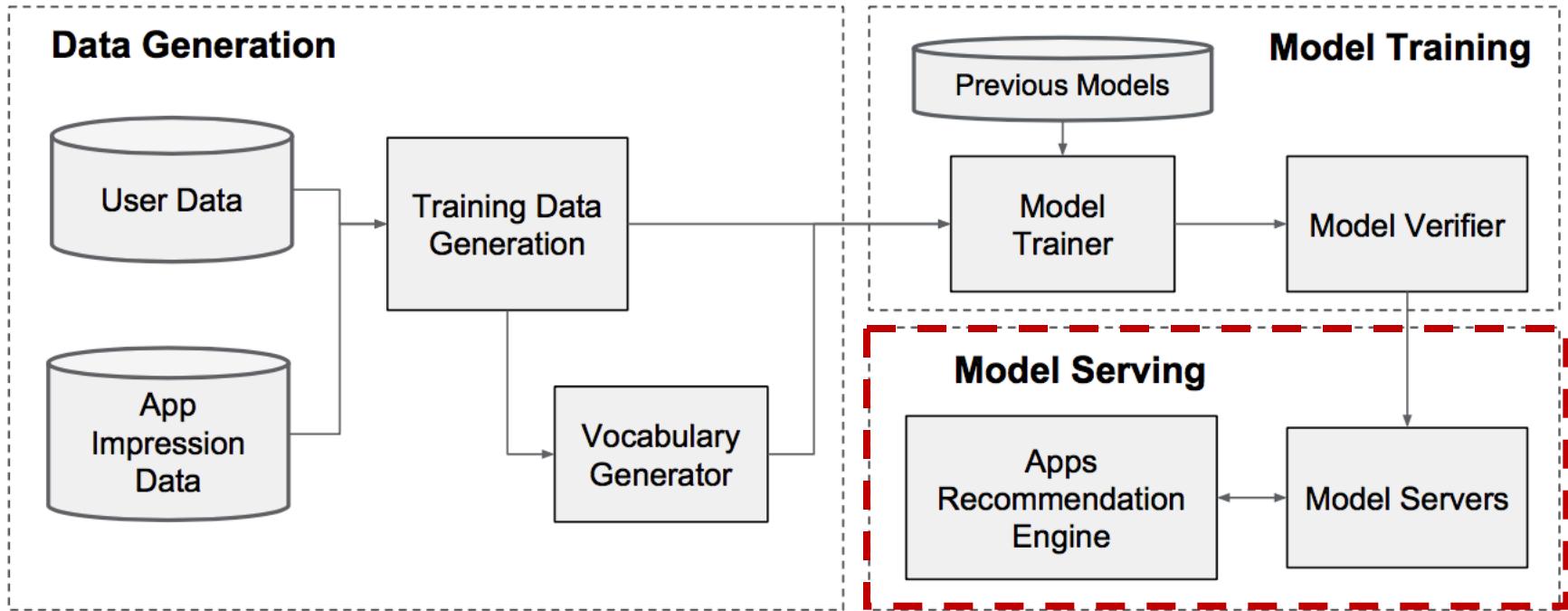




Wide & Deep



Wide & Deep



Deep and Wide Notebook

- Demonstrate how to use the tf.estimator API to train a wide linear model and a deep feed-forward neural network.

bit.ly/strata17recsys

The Goal

- Our Goal is to predict the rating category for a given pair of user id and movie id

gender	age_desc	occ_desc	title	genre	rating	prediction	rating1	rating2	rating3	rating4	rating5	
223742	M	35-44	clerical/admin	Little Princess, The (1939)	Children's Drama	4	4	0.025210	0.068422	0.234015	0.375578	0.296774
915512	M	25-34	unemployed	Misery (1990)	Horror	4	3	0.105347	0.148656	0.290903	0.287796	0.167298
209015	M	25-34	technician/engineer	Supercop (1992)	Action Thriller	3	4	0.050473	0.120541	0.292055	0.350531	0.186400
719570	M	25-34	other or not specified	Red Violin, The (Le Violon rouge) (1998)	Drama Mystery	3	4	0.057457	0.115074	0.279706	0.332208	0.215554
283590	M	35-44	programmer	Manon of the Spring (Manon des sources) (1986)	Drama	4	4	0.025996	0.079624	0.237717	0.393170	0.263494

DnW Accuracy: 34.907003%

Implementation Steps

- Data preprocessing
- Build inputs
- Hash categorical features
- Create embeddings of sparse features for the deep model
- Define features for both the deep and the wide part of the model
- Train and validate the model



Big chunk !

Preparing the inputs

```
model =  
tf.contrib.learn.DNNLinearCombined  
Classifier(.....  
)
```

Agenda

- Recommender Systems 101
- Collaborative Filtering
- Deep & Wide Learning
- Deep Matrix Factorization

Deep Matrix Factorization Notebook

- Demonstrate how to use the tensorflow lower API to build a DNN model from scratch.

bit.ly/strata17recsys

The Goal

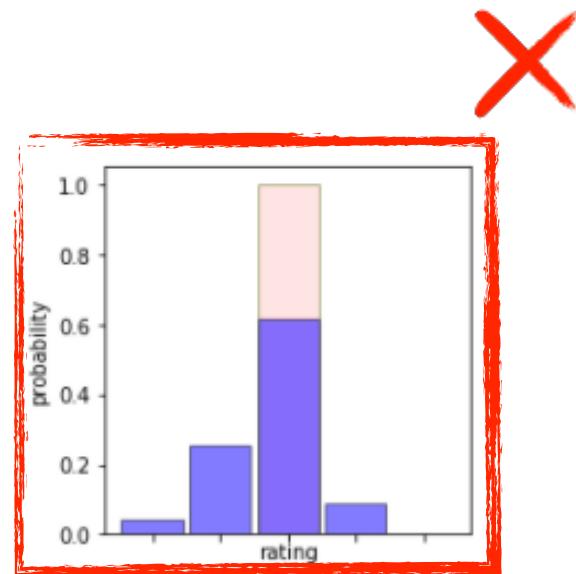
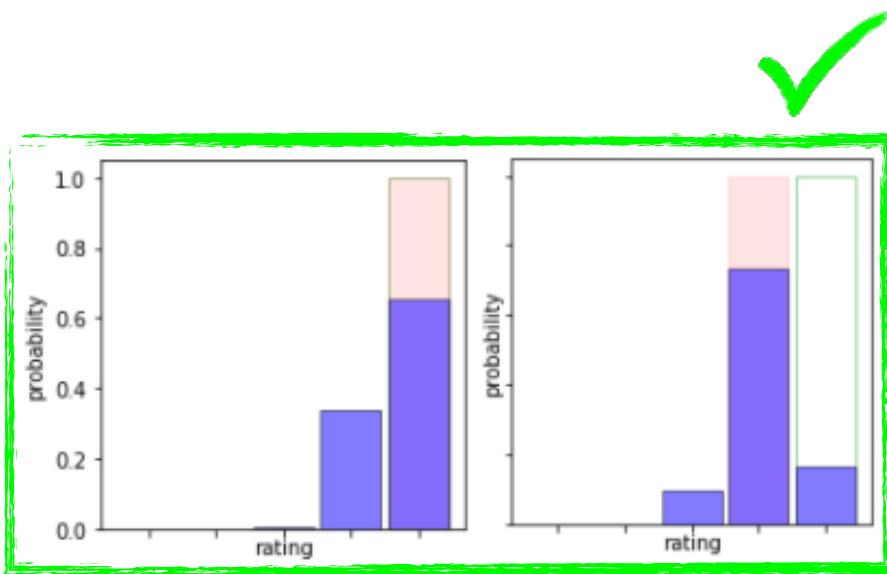
- Our Goal is not only to get a single prediction value, but also to know the probability distribution of the prediction.

gender	age_desc	occ_desc	title	genre	rating	prediction	rating 1	rating 2	rating 3	rating 4	rating 5	
690811	M	35-44	college/grad student	Ravenous (1999)	Drama Horror	4	3	3.967836e-03	0.053077	0.484498	0.420719	0.037738
838395	M	56+	retired	Miss Julie (1999)	Drama	4	3	4.872613e-03	0.055875	0.457305	0.434597	0.047350
228947	M	25-34	sales/marketing	2001: A Space Odyssey (1968)	Drama Mystery Sci-Fi Thriller	5	4	6.138799e-05	0.001535	0.041855	0.509231	0.447319
555611	M	Under 18	K-12 student	Saving Private Ryan (1998)	Action Drama War	5	5	2.225889e-08	0.000003	0.000494	0.102754	0.896749
745274	F	25-34	sales/marketing	Iron Giant, The (1999)	Animation Children's	4	4	1.565472e-06	0.000149	0.016105	0.543688	0.440055

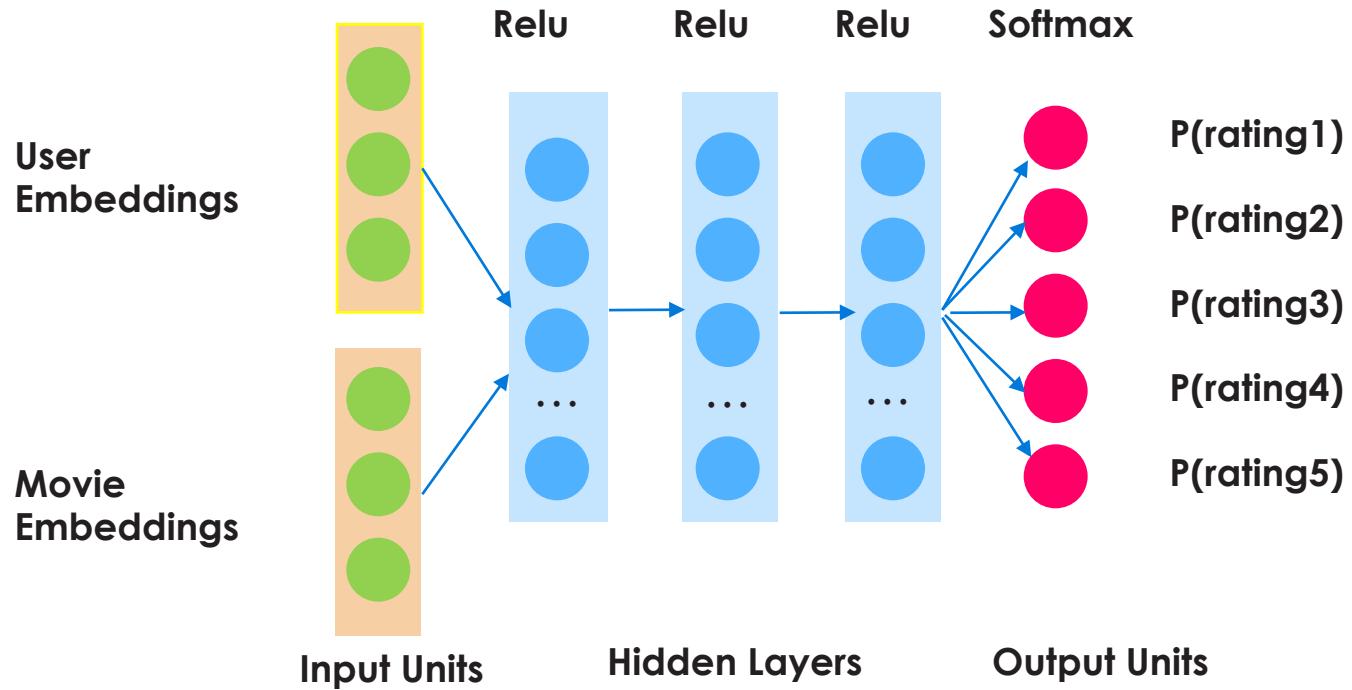
DNN Accuracy: 50.559383%

The Advantage

- With this additional probability information we can assess the risks and certainty of our result.

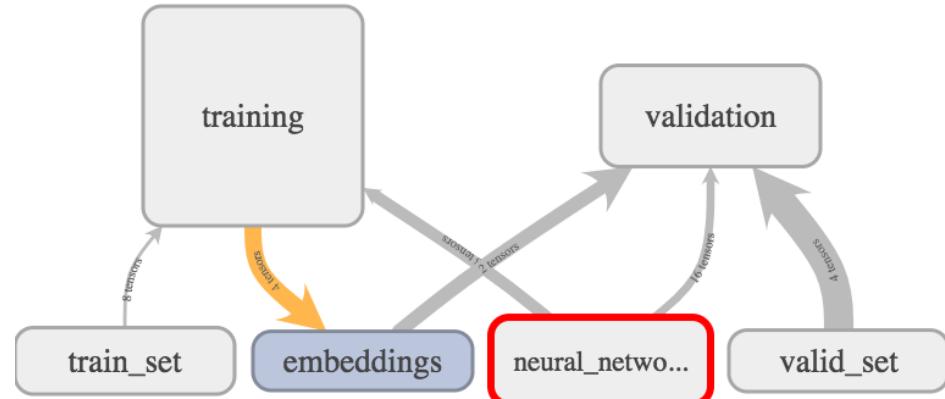


The Model



Implementation Steps

- Initialize the movie and user embeddings with the results trained in MF model
- Initialize the DNN parameter W and b
- Calculate Z n
- Calculate the prediction
- Calculate the loss
- Train and validate the model



`tensorboard --logdir=path/to/log-directory`



A TERADATA COMPANY