

join at Slido.com with #bigdata2018

**BIG DATA
CONFERENCE.**

+ datatonic

Deep Learning for Recommender Systems

Oliver Gindele

@tinyoli

oliver.gindele@datatonic.com

Big Data Conference Vilnius

28.11.2018

Who is Oliver?

- + Head of Machine Learning
- + PhD in computational physics



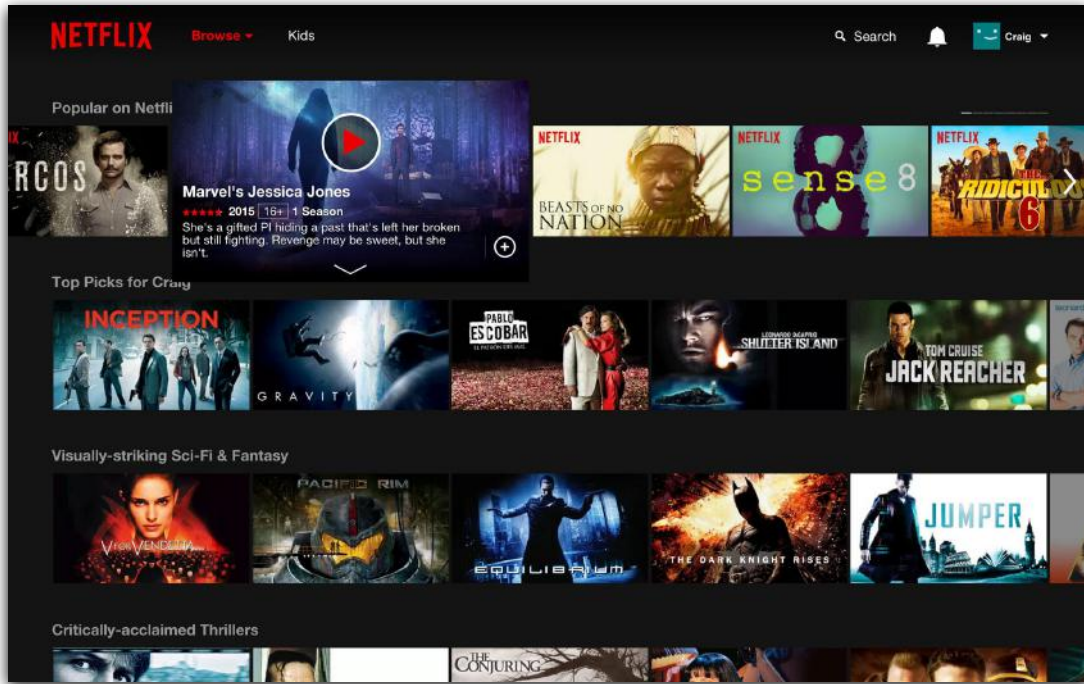
Who is datatonic?

We are a **strong team** of **data scientists**, **machine learning experts**, **software engineers** and **mathematicians**.

Our mission is to **provide tailor-made** systems to help your organization get **smart actionable insights** from **large data volumes**.

+ datatonic

Recommender Systems



Recommender Systems

The image shows a screenshot of the Spotify Discover Weekly playlist. The interface is dark-themed. On the left is a navigation sidebar with sections for 'MAIN' (Browse, Activity, Radio), 'YOUR MUSIC' (Songs, Albums, Artists, Local Files), and 'PLAYLISTS' (Discover Weekly by spo..., Starred, iTunes). The main content area features the 'Discover Weekly' playlist header with a cover image of a dog on a moon. Below the header are 'PLAY' and 'FOLLOWING' buttons, and a 'FOLLOWERS' count. A table lists the songs in the playlist, including 'Strong - Claude VonStroke Remix', 'California', 'Black Jeans', 'All I Want - Diplo Remix', 'A Stranger Love (Salva Remix)', and 'Control'. An 'Available Offline' toggle is visible on the right.

MAIN

- Browse
- Activity
- Radio

YOUR MUSIC

- Songs
- Albums
- Artists
- Local Files

PLAYLISTS

- Discover Weekly by spo...
- Starred
- iTunes

PLAYLIST

Discover Weekly

Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your favorites!

Created by: Spotify · 30 songs, 2 hr 5 min

PLAY FOLLOWING ...

FOLLOWERS 0

Available Offline

SONG	ARTIST	ALBUM	
+ Strong - Claude VonStroke Remix	London Grammar	If You Wait - Remixes 2	3 days ago
+ California	Letch	+	3 days ago
+ Black Jeans	Mayeoni	Black Jeans	3 days ago
+ All I Want - Diplo Remix	Dawn Golden	Still Life	3 days ago
+ A Stranger Love (Salva Remix)	Classbox	A Stranger Love (Remi..	3 days ago
+ Control	Kevin Garrett	Control	3 days ago

Recommender Systems

The image shows a search engine results page for the query 'python'. The search bar at the top contains the text 'python' and has a microphone icon and a search icon. Below the search bar, there are navigation tabs for 'All', 'Images', 'News', 'Videos', 'Books', and 'More', along with 'Settings' and 'Tools'. The search results are displayed in a grid format. On the left side, there are several search results, including 'Welcome to Python.org', 'Python (programming language) - Wikipedia', and 'Python | Codecademy'. On the right side, there is a large, detailed search result for 'Python', which includes a description, a list of key features, and a section for 'People also search for' with icons for Java, C++, JavaScript, PHP, and C.

python

All Images News Videos Books More Settings Tools

About 65,700,000 results (0.44 seconds)

Welcome to Python.org
<https://www.python.org/>
The official home of the Python Programming Language.

Search python.org

Download
Python 3.6.4 - Windows -
Python2orPython3 - Python 3.5.3

Python For Beginners
BeginnersGuide -
BeginnersGuide/Download - IDEs -
Python Editors

The Python Tutorial
1. Whetting Your Appetite - 5. Data
Structures - 9. Classes - ...

Windows
Python Releases for Windows. Latest
Python 3 Release ...

Documentation
Browse the docs online or download a
copy of your own ...

Other Platforms
Download Python for Other Platforms.
Python has been ...

Python (programming language) - Wikipedia
[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using ...
History · Features and philosophy · Syntax and semantics · Implementations

Python | Codecademy
<https://www.codecademy.com/en/tracks/python>
Learn to program in Python, a powerful language used by sites like YouTube and Dropbox.

Learn Python | Codecademy

python
High-level programming language

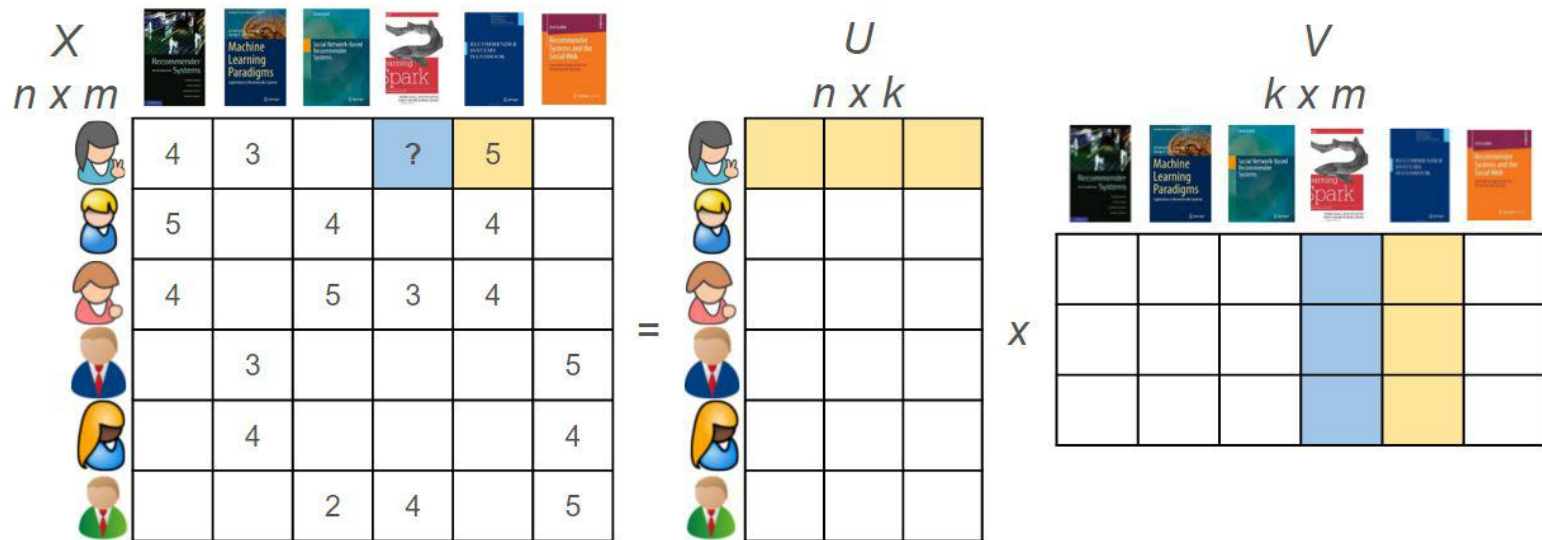
Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code ...
Wikipedia

Typing discipline: Duck, dynamic, strong
Designed by: Guido van Rossum
First appeared: 20 February 1991; 26 years ago
Preview release: 3.7.0b1 / 2018
Stable release: 3.6.4 / 19 December 2017; 49 days ago; 2.7.14 / 16 September 2017; 4 months ago
Filename extensions: .py, .pyc, .pyd, .pyo (prior to 3.5), .pyw, .pyz (since 3.5)

People also search for View 15+ more

Java C++ JavaScript PHP C

Collaborative Filtering - Introduction



Collaborative Filtering - Introduction

Objective:

$$\min_{x_*, y_*} \sum_{r_{u,i} \text{ is known}} (r_{ui} - x_u^T y_i)^2 + \lambda(\|x_u\|^2 + \|y_i\|^2)$$

- + Netflix Prize (2009)
- + Solve via SVD (ALS or SGD)
- + Regression problem

Finding Love with Numbers

Online Dating Dataset – LibimSeTi

The screenshot shows the LibimSeTi website interface. At the top left is the logo "libim.se ti. cz" in red and black. Below it, it says "640 online, přihlášeno 123 žen, 255 mužů". To the right is a search bar with a "vyhledat" button. A navigation menu includes: "MOJE LÍBIMSETI | HODNOCENÍ | SEZNAMKA | UŽIVATELE | FOTKY | LÍBIMSETI LIFE | DISKUZE | CHAT | MISS | HRY | VIDEO | DALŠÍ".

On the left side, there is a blue "Přihlášení" (Login) box with fields for email and password, and a "přihlásit" button. Below it is a pink box "Proč být na Líbimseti" with a list of features: "nejoblíbenější zábavní portál", "vlastní profil, fotoalba", "unikátní chat a e-mail", "klubové akce, soutěže", and "oblíbený server celebrit". Below that is a yellow "Změnit vzhled" (Change appearance) box with a dropdown menu showing "Libimseti.cz" and a "»" button.

The main content area is divided into two sections: "Náhodný výběr žen (změnit)" and "Náhodný výběr mužů (změnit)".

Náhodný výběr žen (změnit):

- Wewerca-Cofola194** (23.8): Profile picture of a woman, location "Praha-východ".
- kockalenka22** (23.5): Profile picture of a woman with a car, location "Plzeňský kraj".
- Kismeassbaby** (20.0): Profile picture of a woman, location "Litoměřice".

Náhodný výběr mužů (změnit):

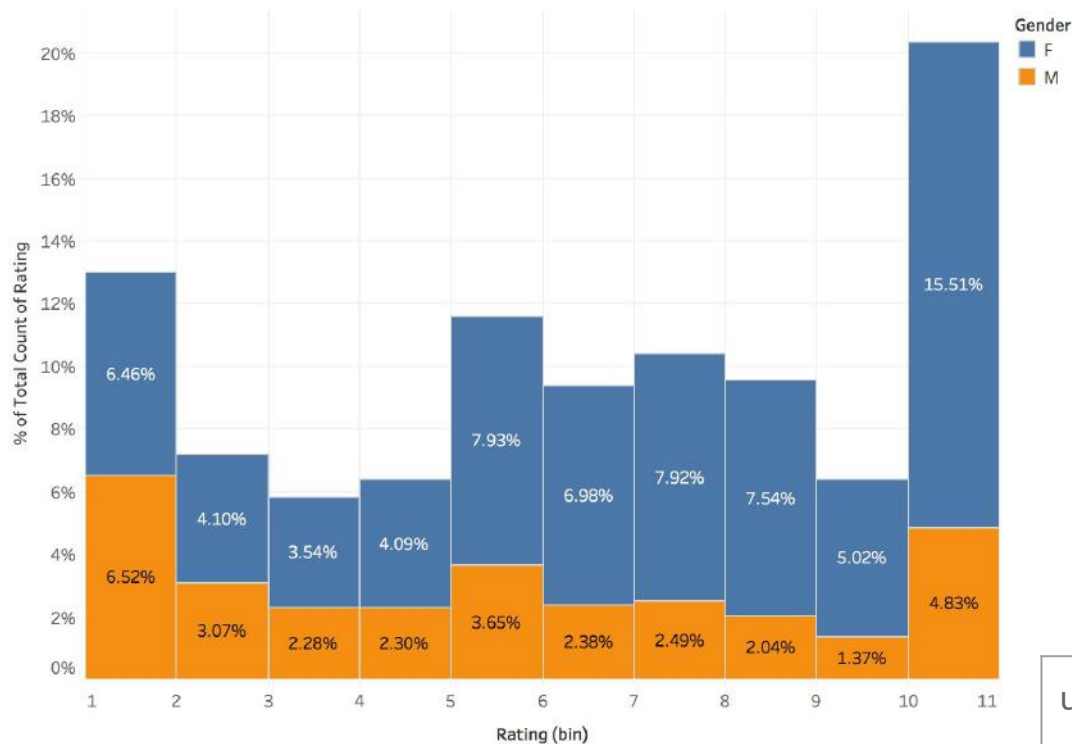
- martinchra** (24.7): Profile picture of a man, location "Praha 5".
- Davidek_21** (22.4): Profile picture of a man, location "Vysocina".
- BenSolo** (22.4): Profile picture of a man, location "České Budějovice".

At the bottom of the main content area is a large blue button with the text "Poprvé na Líbimseti.cz ?" and a right-pointing arrow. Below the button is a "Chat" link.

On the right side, there is a "MAGAZIN" section with several article thumbnails and titles:

- "Muž jako oběť domácího násilí"
- "Velký návrat šponážních románů"
- "To nehorší, co si můžete vzít na sebe"
- "Moderní nábytek do dětského pokoje splní tv nejtajnější sny vašich ratolestí"
- "Které slovánské ženy dneška jsou nehezčí?"
- "Jak stáhnout hudbu z Youtube do formátu MP3"
- "Originální korálkové náramky: Trend, jemuž neodoláte"
- "Vimax: Efektivní a ekonomické řešení problémů s erekcí"
- "Jaká jsou nebezpečnější místa v Evropě pro cestování a turistiku?"
- "Práce z domu - pohoda, či skryté nepřítomnosti?"

Online Dating Dataset – LibimSeTi



+ <http://www.libimseti.cz/>

+ 2005

+ 17,359,346 ratings

+ 135,359 users

+ Ratings: 1-10

+ Female (%): 69

+ Male (%): 31

+ Mean(rating): 5.9

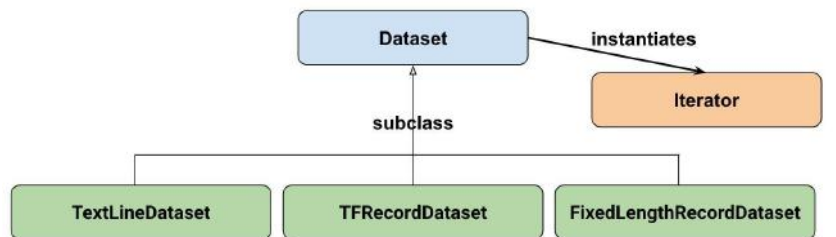
+ Std(rating): 3.1

userId	profileId	rating	gender
--------	-----------	--------	--------

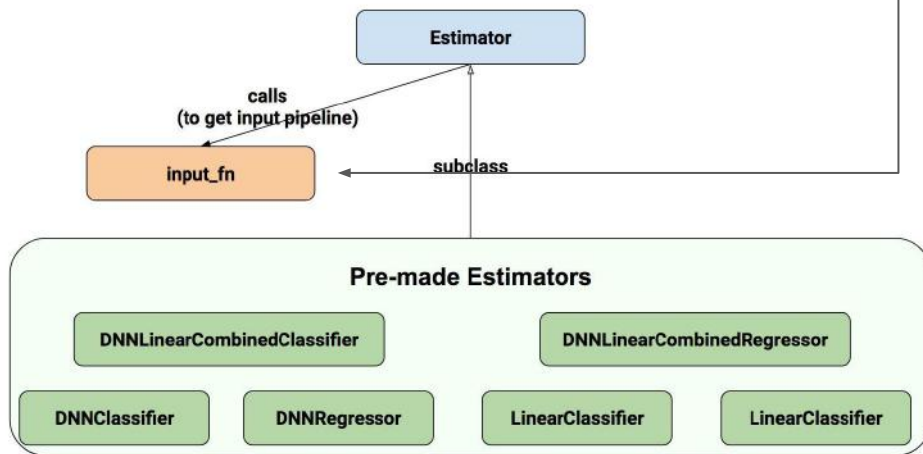
Tensorflow: High Level APIs



Dataset:



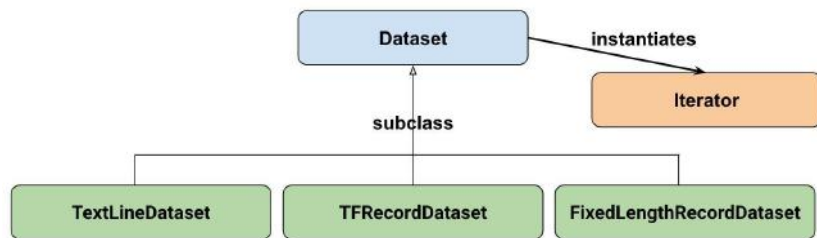
Estimator:



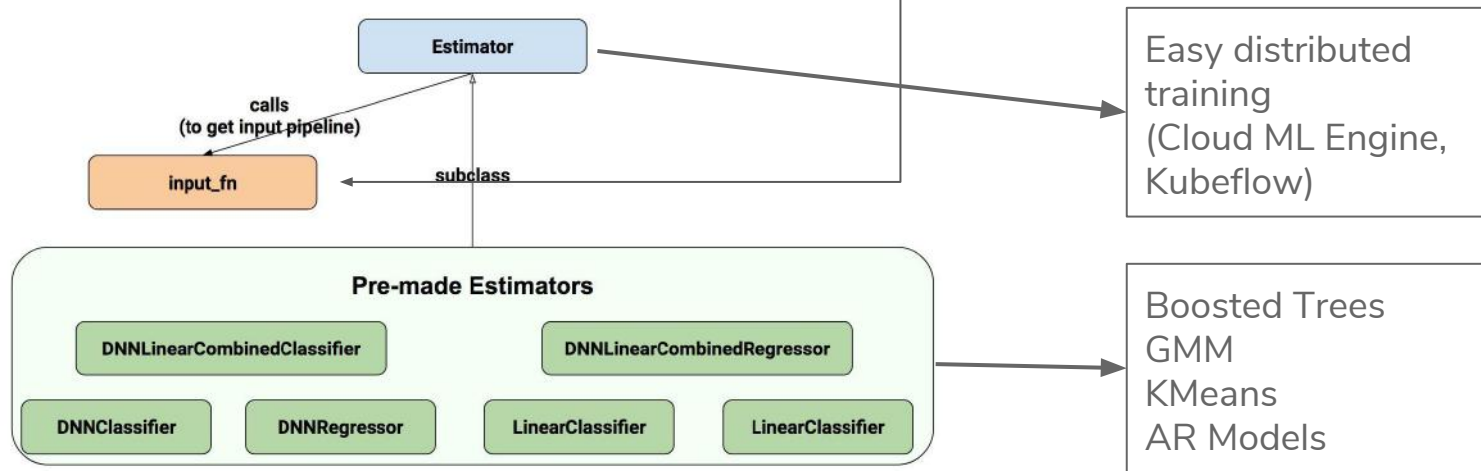
Tensorflow: High Level APIs



Dataset:



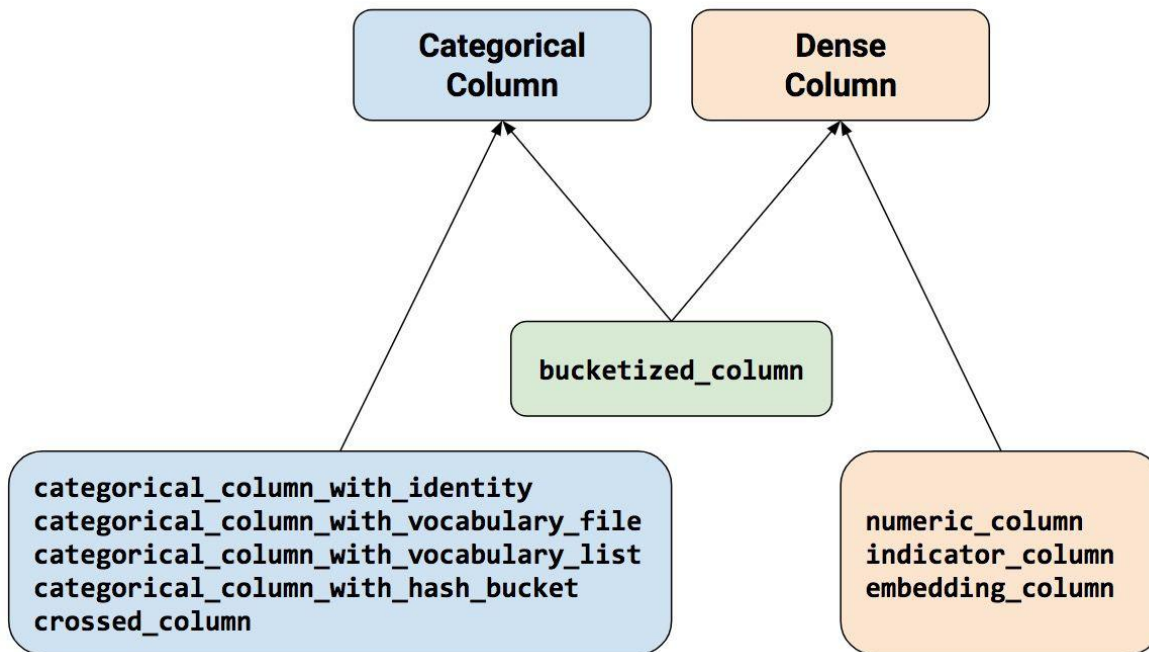
Estimator:



Tensorflow: High Level APIs



Feature Columns:



Dataset API (tf.data)

```
def input_fn(data_file, num_epochs=None, shuffle=True, batch_size=512, skip_header_lines=1):
    """Generate an input function for the Estimator."""
    def parse_csv(value):
        print('Parsing', data_file)
        columns = tf.decode_csv(value, record_defaults=CSV_COLUMN_DEFAULTS)
        features = dict(zip(COLUMNS, columns))
        labels = features.pop(LABEL_COLUMN)
        return features, labels

    # Extract lines from input files using the Dataset API.
    dataset = tf.data.TextLineDataset(data_file).skip(skip_header_lines)
    if shuffle:
        dataset = dataset.shuffle(buffer_size=10000)

    dataset = dataset.apply(tf.contrib.data.map_and_batch(parse_csv, batch_size, num_parallel_batches=1))
    dataset = dataset.cache()
    dataset = dataset.repeat(num_epochs)
    dataset = dataset.prefetch(1)

    iterator = dataset.make_one_shot_iterator()
    features, labels = iterator.get_next()

    return features, labels
```

MF Model

```
userid = tf.feature_column.categorical_column_with_hash_bucket("userid", hash_bucket_size=10000)
profileid = tf.feature_column.categorical_column_with_hash_bucket("profileid", hash_bucket_size=10000)
columns = [userid, profileid]
mf_feature_columns = [tf.feature_column.embedding_column(x, dimension=10) for x in columns]
mf_bias_columns = [tf.feature_column.embedding_column(x, dimension=1) for x in columns]

tensors = tf.feature_column.input_layer(features, mf_feature_columns)
biases = tf.feature_column.input_layer(features, mf_bias_columns)

userid, profileid = tf.split(tensors, 2, axis=1)
bias_userid, bias_profileid = tf.split(biases, 2, axis=1)
```

```
with tf.device(params['device']):
    # Calculate dot product
    model = tf.reduce_sum(tf.multiply(userid, profileid), 1)
    # Add biases
    model = tf.add(model, tf.squeeze(bias_userid, axis=1))
    model = tf.add(model, tf.squeeze(bias_profileid, axis=1))

    # Add regularization
    l2_reg = tf.contrib.layers.apply_regularization(
        tf.contrib.layers.l2_regularizer(params['l2_beta_MF'], scope="l2_reg"), weights_list=[userid, profileid])

    # Calculate loss using mean squared error
    loss = tf.losses.mean_squared_error(labels, model)
    loss = tf.add(loss, l2_reg)
```

Estimator API (tf.estimator)

```
# Calculate metrics
eval_metric_ops = {
    "rmse":
        tf.metrics.root_mean_squared_error(labels, model),
    "mae":
        tf.metrics.mean_absolute_error(labels, model),
}

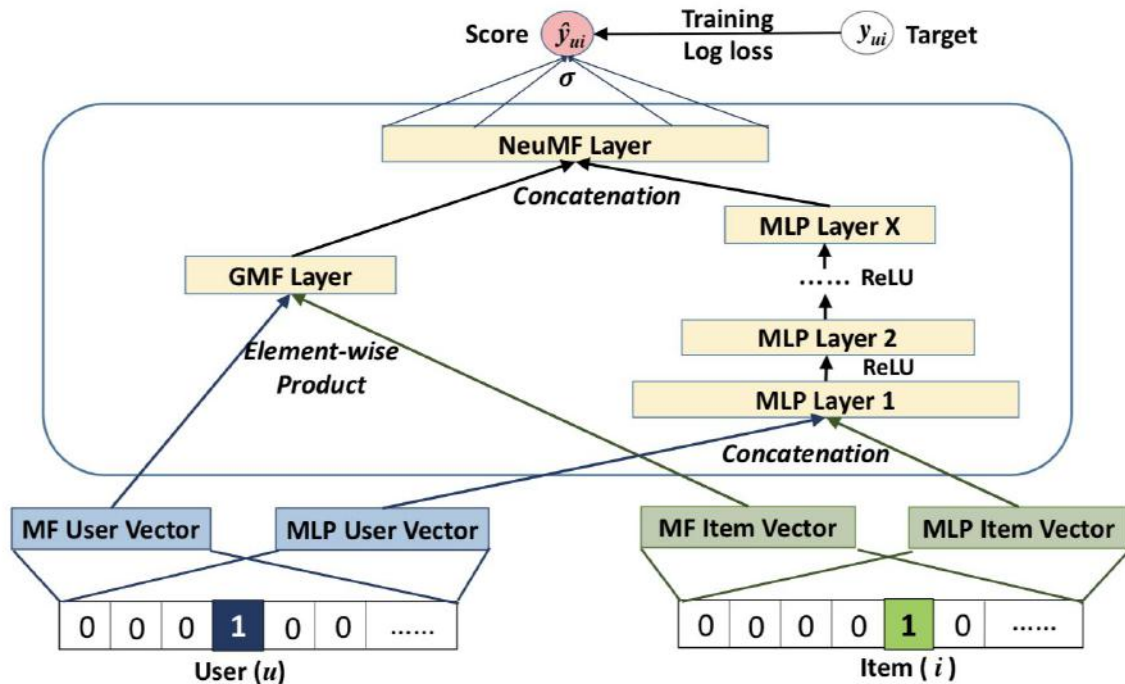
train_op = tf.contrib.layers.optimize_loss(
    loss=loss, global_step=tf.train.get_global_step(),
    learning_rate=0.001, optimizer='Adam')

model_fn = tf.estimator.EstimatorSpec(mode=mode, predictions=predictions_dict, loss=loss,
    eval_metric_ops=eval_metric_ops, train_op=train_op)

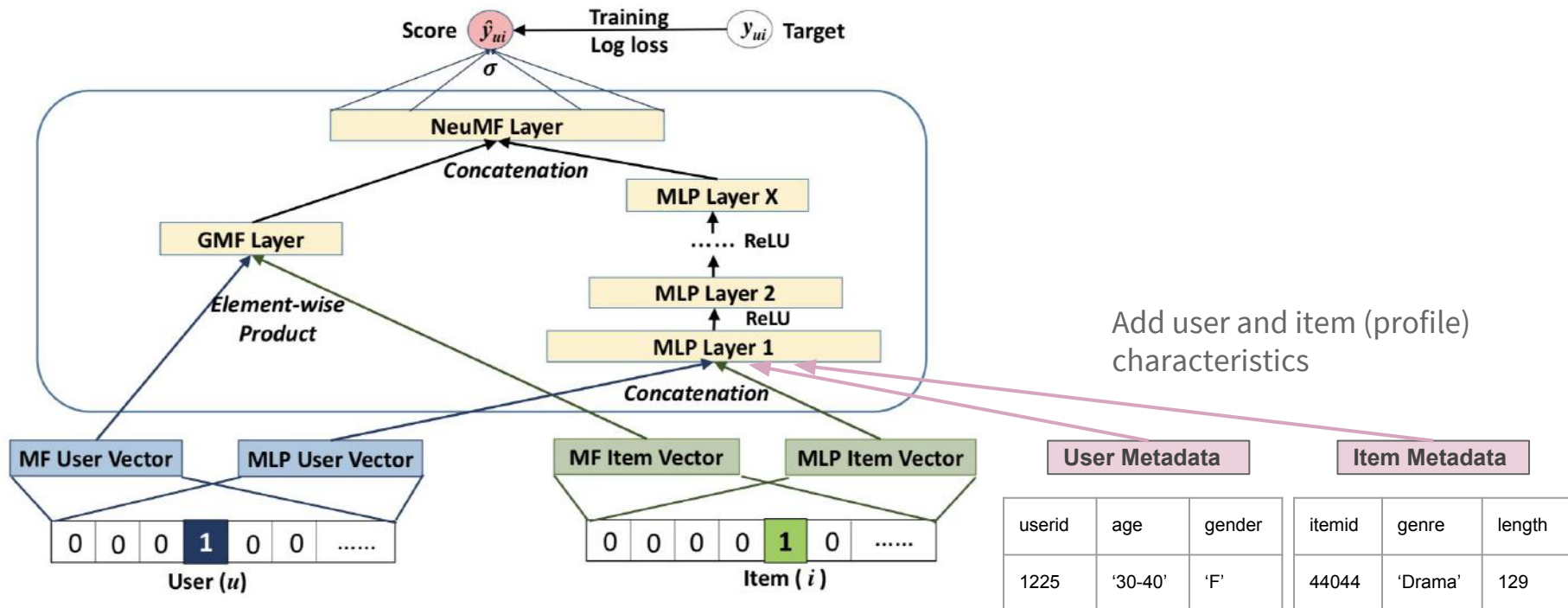
model = tf.estimator.Estimator(model_fn=model_fn, params=model_params, config=run_config, model_dir=model_dir)
```


Going Deeper - Beyond MF

Neural Collaborative Filtering (He et al.)



Going Deeper - Beyond MF



Results - LibimSeTi

	MF	MLP	MF + MLP	Research [1]
RMSE	2.137	2.112	2.071	2.077
MAE	1.552	1.541	1.432	1.410

Training details:

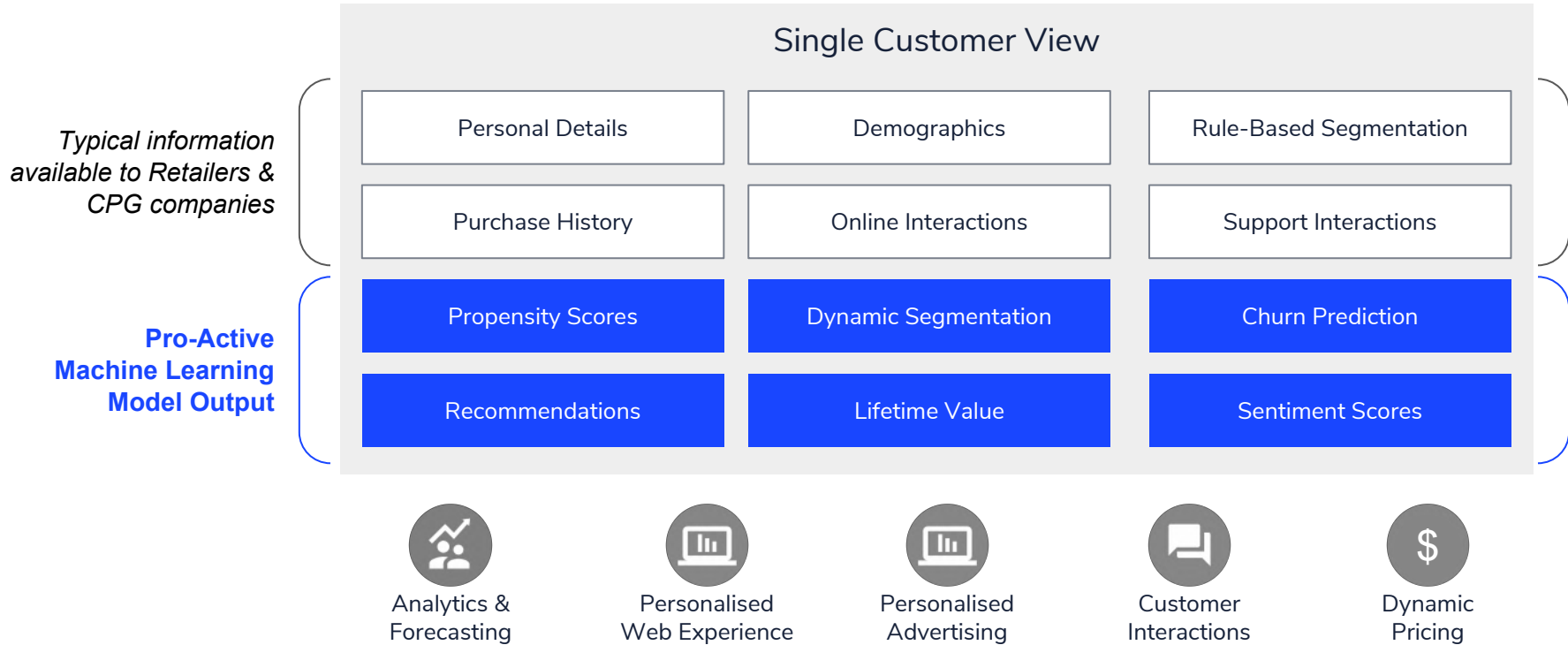
- + 40 epochs
- + MLP: 4 layers (256 units pyramid)
- + Adam optimiser
- + Results calculated on held out test set (5 rating per user)
- + No tuning of hyperparameters

[1] Trust-Based Recommendation: an Empirical Analysis, O'Doherty, Jouili, Van Roy (2008)

How can we do better?

Better/More Data

Better/More Data



Better Loss Functions

Better Loss functions

- + Implicit feedback (Hu, Koren, Volinsky 2008):

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

- + Logistic Matrix Factorisation (Johnson, Spotify, 2014):

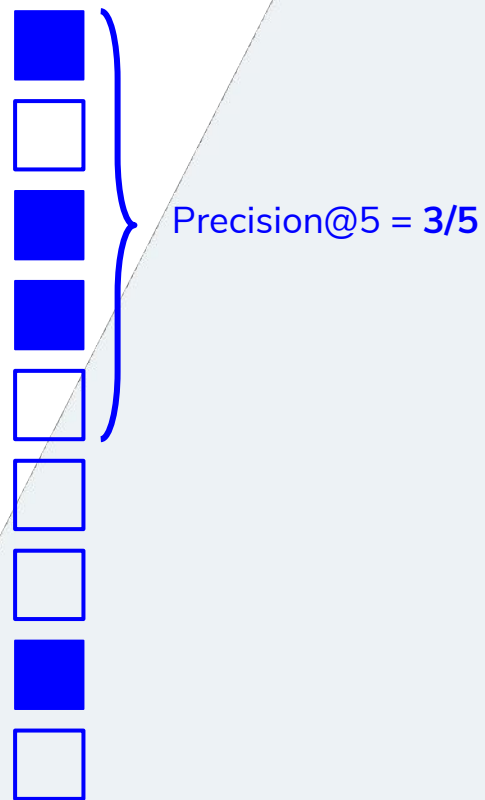
$$p(l_{ui} | x_u, y_i, \beta_i, \beta_j) = \frac{\exp(x_i y_i^T + \beta_u + \beta_i)}{1 + \exp(x_u y_i^T + \beta_u + \beta_i)}$$

- + Use ranking loss (or pairwise loss functions)

Improved Metrics

Precision@k



















Of the top k recommendations, how many were relevant (interacted with)?



(Normalized) Discounted Cumulative Gain

DCG given by formula

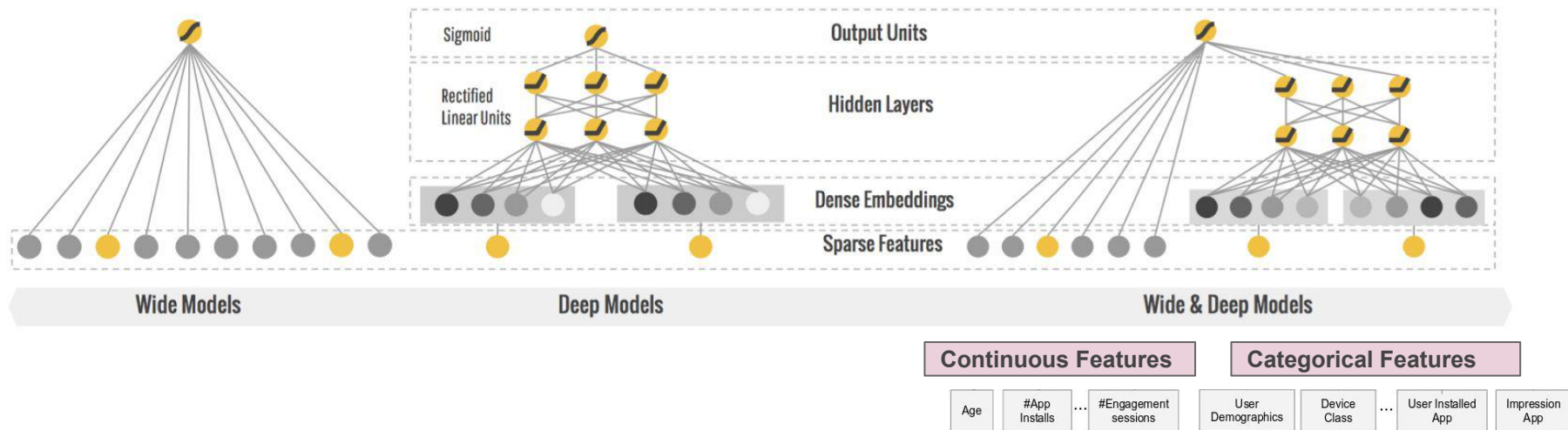
$$\sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$$

		1	$1 / \log_2(1+1)$
		2	
		3	$1 / \log_2(3+1)$
		4	$1 / \log_2(4+1)$
		5	
		6	
		7	
		8	$1 / \log_2(8+1)$
		9	
		DCG	= $\frac{2.246}{2.562} = 0.877$
		Best DCG	= 2.562

Go even deeper and embed
all the things!

Deep Recommender Systems – Advances

Wide & Deep model (Cheng et al., 2016)



Results - Real Client Data

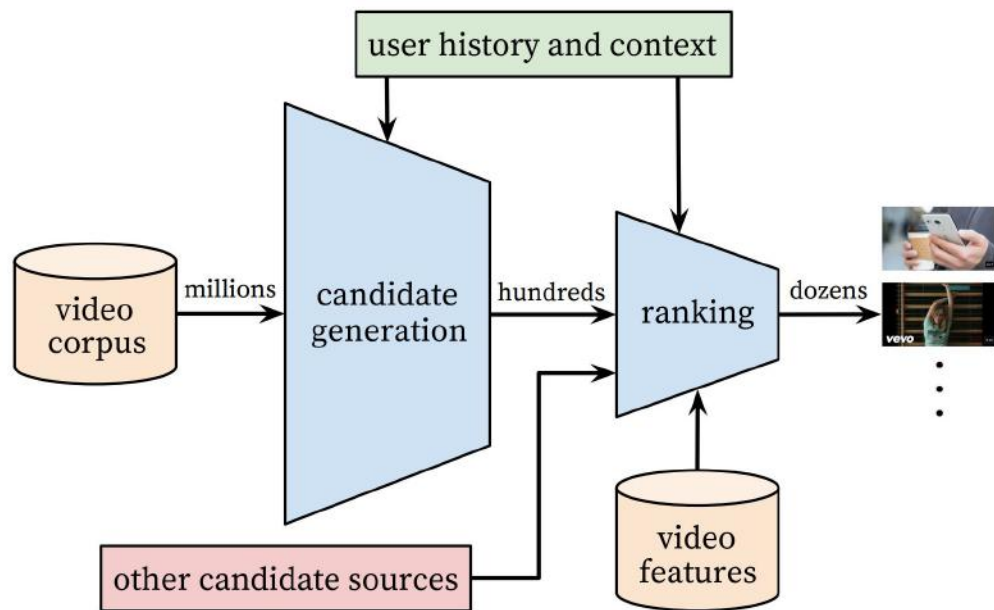
	SVD (MF)	MF - Implicit Feedback model	Wide & Deep
P@5	0.33	0.51	0.79
NDCG@5	0.18	0.30	0.37

Training details:

- + Feature columns include user demographics to complement the lack of interactions of cold users
- + 100 epochs
- + Adam optimiser
- + Results calculated on held out test set (up to 5 ratings per user)
- + Tuning of the dimension size of the embedding vector of the deep part

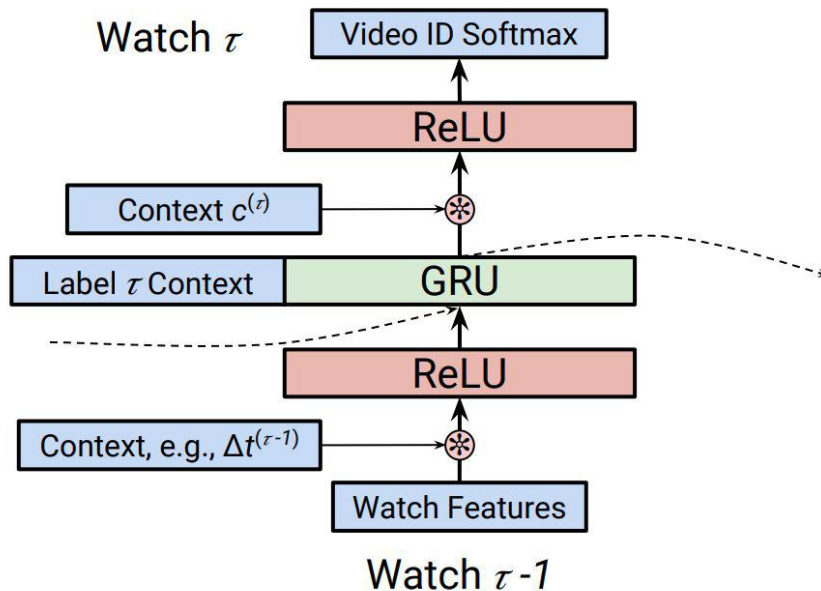
Deep Recommender Systems - Advances

Deep Neural Networks for YouTube Recommendations (Covington, Adams, Sargi, 2016)



Deep Recommender Systems - Advances

Latent Cross: Making Use of Context in Recurrent Recommender Systems (Alex Beutel, Paul Covington, Sagar Jain et al. , 2018)



Takeaways

Takeaways

- + Tensorflow can do more than vision or translation
- + High level APIs make model building and training painless
- + Custom algorithms and specific loss functions are easily implemented
- + Deep Recommender systems work well on real data
- + Embeddings and hidden layers allow for many ways to improve a recommender system

Thank you.



[Blog.datatonic.com](https://blog.datatonic.com)



facebook.com/datatonic



linkedin.com/company/datatonic



twitter.com/teamdatatonic