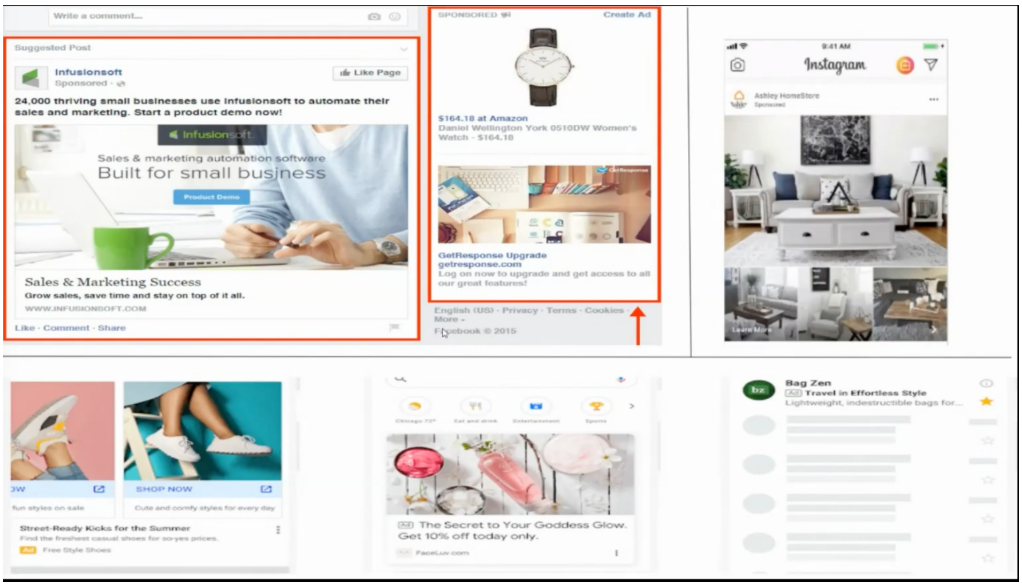
ML System Design - Ad Click Predictions on Social Platforms

Video: https://www.youtube.com/watch?v=wxUx3gIUEsk

Business Objective:

Ad Clicks Prediction on Social Media Platform

Ads on Different Platforms

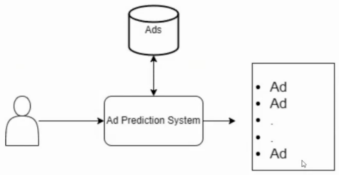


Clarifying requirements:

- To design an Ad Prediction System;
  - Business Objective - To maximize revenue.
- Assumptions and Constraints:
  - Ads on user feed.
  - No “Ad fatigue period” and “Block Advertiser” but have “hide this ad”.

Framing the problem as an ML task: To correctly predict if the user will click the ad.

- Specifying input/output:



Data Preparation : Data Engineering: 1. Ad, 2. User, 3. User-Ad Interaction

Ad data:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0
5	59.99	23	59761.56	226.74	Sharable client-driven software	Jamieberg	1	Norway	2016-05-19 14:30:17	0
6	88.91	33	53852.85	208.36	Enhanced dedicated support	Brandonstad	0	Myanmar	2016-01-28 20:59:32	0
7	66.00	48	24593.33	131.76	Reactive local challenge	Port Jefferybury	1	Australia	2016-03-07 01:40:15	1
8	74.53	30	68862.00	221.51	Configurable coherent function	West Colin	1	Grenada	2016-04-18 09:33:42	0
9	69.88	20	55642.32	183.82	Mandatory homogeneous architecture	Ramirezton	1	Ghana	2016-07-11 01:42:51	0

User:

ID	Username	Age	Gender	City	Country	Language	Timezone

User Ad interaction data:

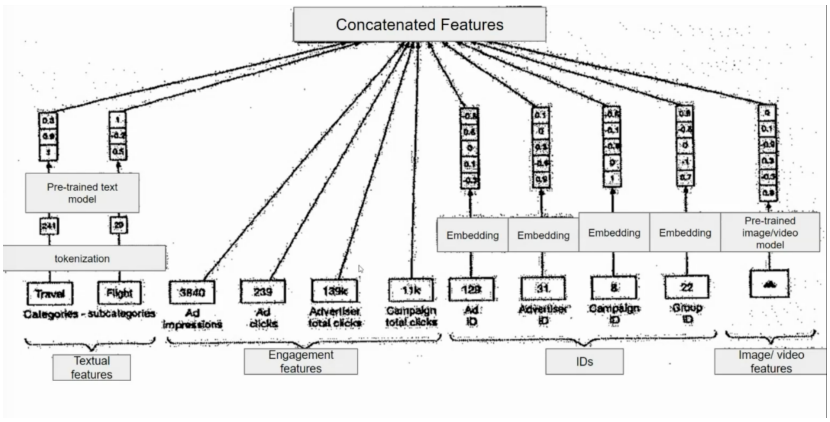
User ID	Ad ID	Interaction type	Dwell time[1]	Location (lat, long)	Timestamp
11	6	Impression	5 sec	38.8951 -77.0364	165845053
11	7	Impression	0.4 sec	41.9241 -89.0389	1658451365
4	20	Click	-	22.7531 47.9642	1658435948
11	6	Conversion	-	22.7531 47.9642	1658451849

Dwell time - total time an ad is present on user's screen.

Data Preparation:

Feature Engineering:

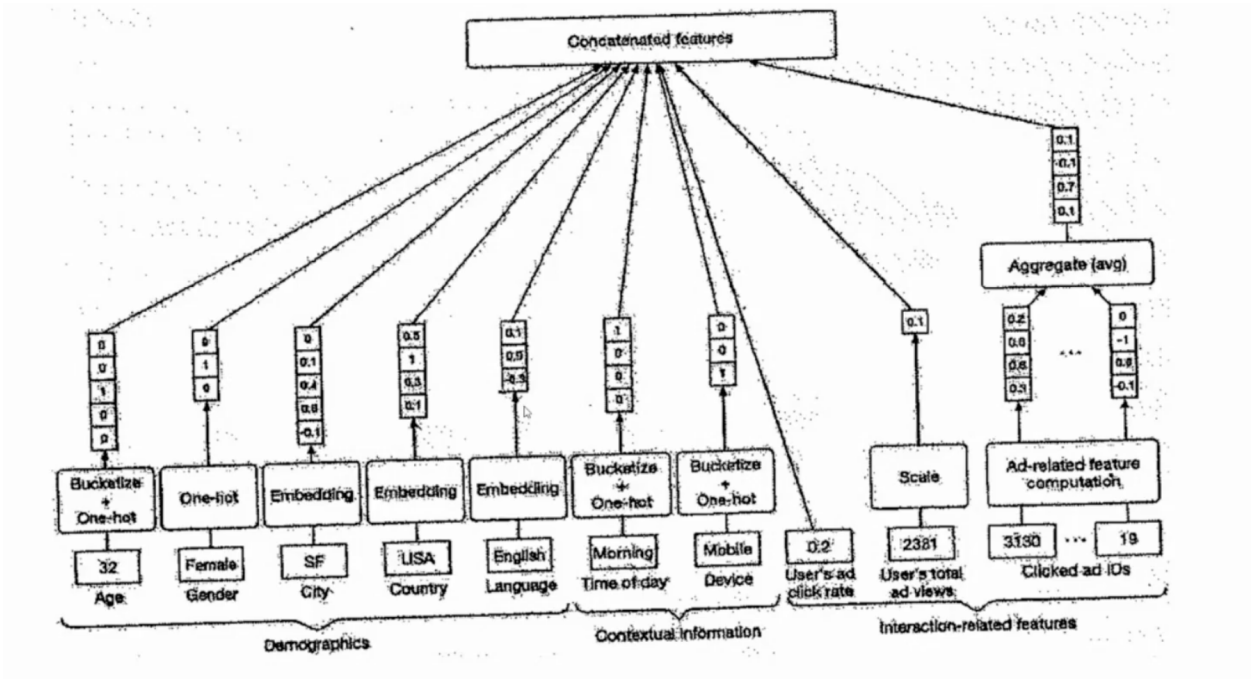
- ID
- Image/Video
- Category and Subcategory
- Impression and click numbers
  - Total impressions/clicks on ad
  - Total impressions /clicks on ads supplied by an advertiser.
  - Total impressions of the campaign.



Data Preparation: Feature Engineering

User Features:

- Demographics : age, gender, city, country etc.
- Contextual Information : device, time of the day etc.
- Interaction-related features :
  - Clicked ads.
  - User's historical engagement.

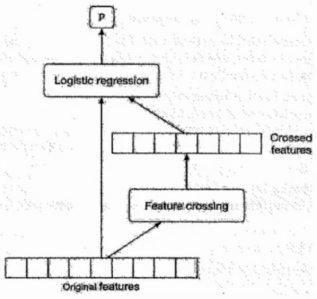


Model Development: Model Selection

- Logistic Regression (LR).
  - Non-linear problems can't be solved with linear decision boundaries.
  - Inability to capture feature interactions.

- Feature Crossing + LR.
  - f1: Country : [USA, China,England]
  - f2: Languages : [English, Chinese]

Country X Languages → f3 : USA and English, f4 : USA and Chinese, f5 : England and English, f6 : England and Chinese, f7 : China and English, f8 : China and Chinese

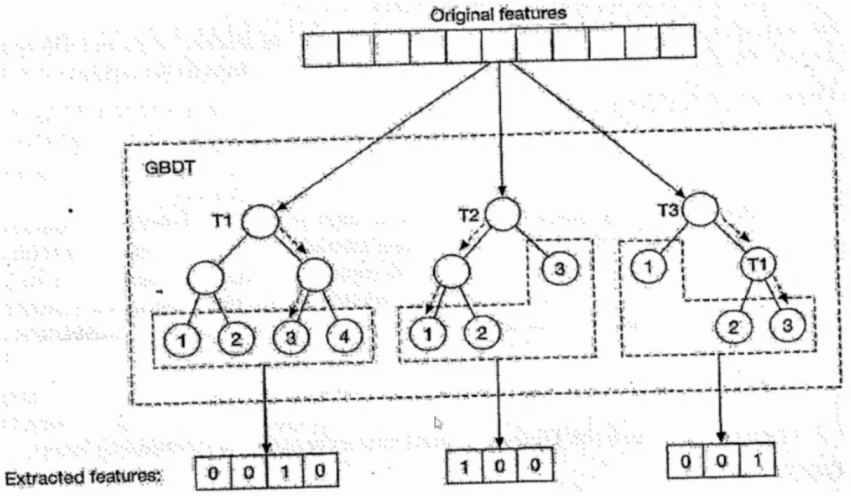


FC + LR Shortcomings:

- Manual Process.
- Requires domain knowledge.
- Cannot process complex interactions.
- Sparsity.

Model Development : Model Selection

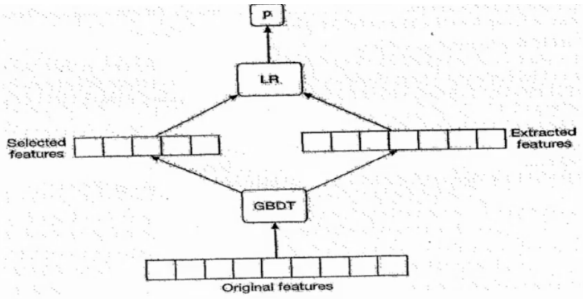
- Gradient Boosted Decision Trees.
  - Pros : interpretable and easy to understand.
  - Cons :
    - Inefficient for continual learning.
    - Cannot train embedding layers.
- Gradient Boosted Decision Tree + Logistic Regression.
  - 1. To train the GBDT model to learn the task.
  - 2. Use feature selection and feature extraction and serve them as input to to LR model for predicting clicks.



Feature Extraction using GBDT.

Feeding GBDT Features into a Logistic Regression Model:

1. **Extract Features:** For each data point, traverse each tree in the GBDT ensemble. At each node, record the decision made (e.g., left or right).
2. **Create Feature Vector:** Concatenate the decisions made at each node for all trees to form a feature vector for the data point.
3. **Feed to Logistic Regression:** Use these new feature vectors as input to a logistic regression model. The logistic regression model can then learn to classify data points based on these GBDT-derived features.



Overview to GBDT + LR.

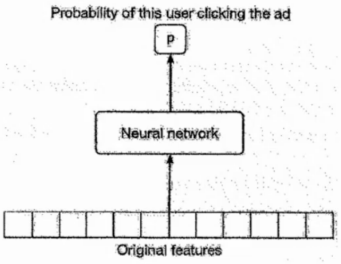
Pros:

- The features produced by GBDT have more predictive power making it easier for LR model to learn the task.

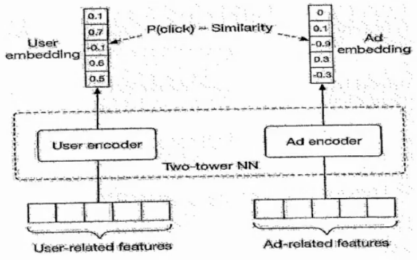
Cons:

- Cannot capture complex interactions - pairwise feature interaction.
- Continual learning is slow.

Model Development : Model Selection.
Neural Networks



1. Single NN



2. Two Tower NN.

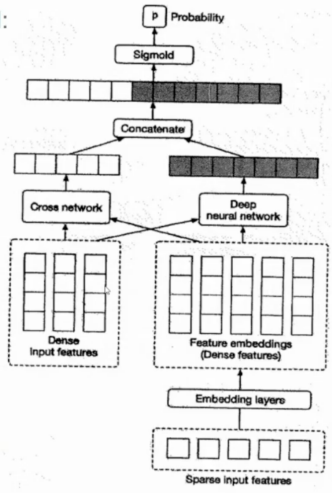
Still not the best Choice because…

- Sparsity.
- Difficult to capture all pairwise feature interactions due to large number of features.

Deep & Cross Network (DCN):

- Deep Network - learns complex and generalizable features using deep neural network.
- Cross Network - Automatically captures feature interactions and learns good features.

Architecture of parallel DCN:



illustrates the architecture of a **Parallel Deep Crossing Network (Parallel DCN)**, which is a deep learning model commonly used for click-through rate (CTR) prediction in online advertising.

Here's a breakdown of the architecture:

Input:

- **Sparse Features:** Categorical features like user ID, product category, etc., which are typically represented as high-dimensional sparse vectors.
- **Dense Features:** Numerical features like age, income, etc., which are usually represented as dense vectors.

Feature Processing:

- **Sparse Features:** Sparse features are fed into embedding layers, which map them to dense vectors of a fixed size. This allows the model to capture the semantic meaning of the categorical features.
- **Dense Features:** Dense features are directly fed into the subsequent layers.

Model Structure:

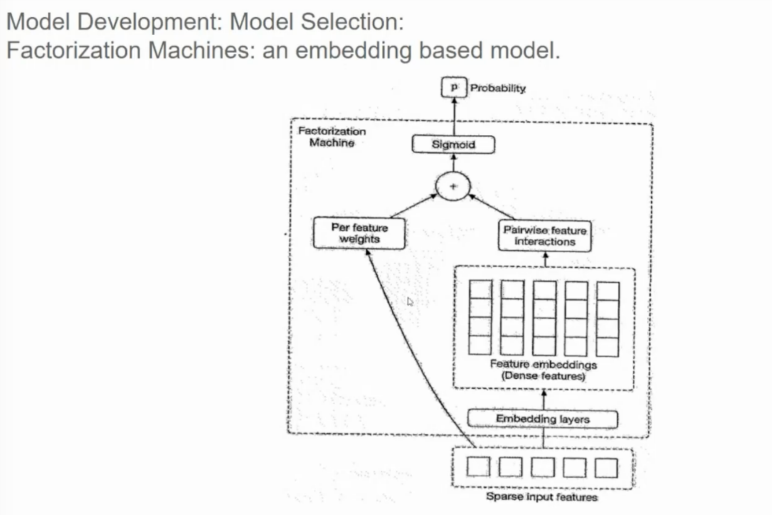
- **Deep Neural Network (DNN):** Both the embedded sparse features and dense features are fed into a deep neural network (DNN) to learn complex non-linear relationships between features and the target variable (click probability).
- **Cross Network:** The cross network is a novel component that introduces interactions between different features. It works by iteratively combining features from different layers, allowing the model to capture higher-order interactions.

Output:

- The output of the DNN and cross network is concatenated and fed into a sigmoid activation function to produce the predicted click probability.

Key Points:

- The parallel structure allows the DNN and cross network to learn different aspects of the data and complement each other.
- The cross network is particularly effective at capturing feature interactions, which are crucial for accurate CTR prediction.
- The model can be trained using techniques like stochastic gradient descent and backpropagation.



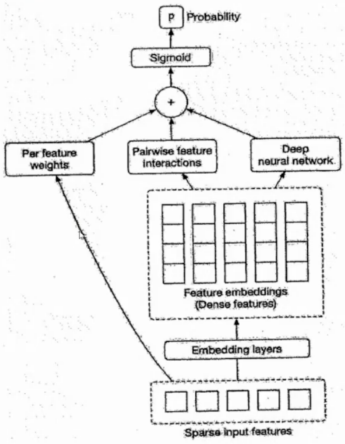
The interaction between two features is determined by dot product of their embeddings.

$$\hat{y}(x) = w_0 + \sum_i w_i x_i + \sum_i \sum_j \langle v_i, v_j \rangle x_i x_j$$

Logistic regression Pairwise interaction

- First 2 terms computes the linear combination of features, similar to Logistic Regression.
- Third term computes pairwise feature interaction.

Deep Factorization Machines (DeepFM):



Alternate way:

- Combining GBDT and DeepFM.

Model Training : Constructing the dataset.

- Input features are computed from the user and ad, depending on that the following labels will be assigned:
  - Positive label - if the user clicks on the ad in less than t seconds.
  - Negative label - if the user does not click the ad in less than t seconds.

#	User and interaction features	Ad features	Label
1	1 0 1 0.8 0.1 1 0	0 1 1 0.4 0.9 0	Positive
2	1 1 0 -0.6 0.9 1 1	1 1 0 0.2 0.7 1	Negative

- The model should be continuously trained in order to adapt new data. So new data points should be continuously generated using new interactions.
- We will us cross entropy as a classification loss function.

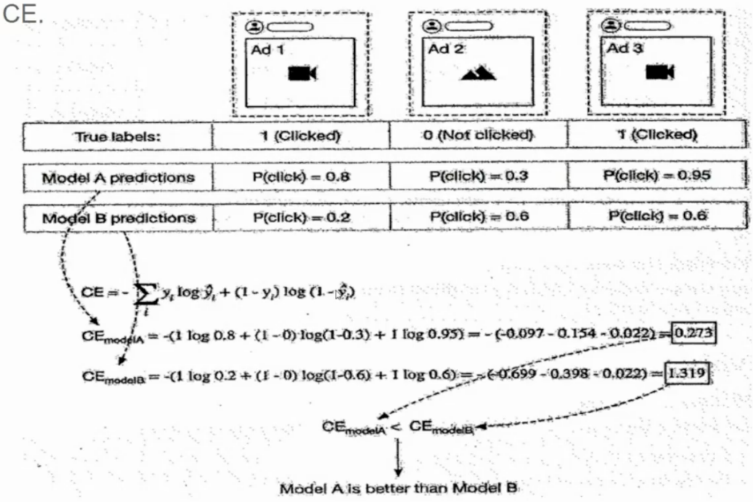
Evaluation Metrics : Offline Evaluation:
Two metrics typically used to evaluate for ad prediction system:

- Cross entropy
- Normalized cross-entropy (NCE)

- Cross entropy - in ideal system we have CE as 0 for negative classed and 1 for positive classes. Lower the CE higher the accuracy.of the prediction.

$$H(p, q) = -\sum_i p_i \log q_i = -\sum_i \left(y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \right)$$

Eg. for CE.



- Normalized Cross Entropy: is ratio of our model's CE and the CE of background average CTR of training data.

$$\text{Normalized cross entropy} = \frac{CE(\text{ML model})}{CE(\text{Simple baseline})}$$

- So, a low NCE indicates model outperforms the simple baseline and an NCE >= 1 shows that model is not performing better than simple baseline.

Online Metrics:

- CTR - number of clicked ads / number of ads shown
- Conversion rate - number of conversions / number of impressions
- Revenue Lift - percentage of revenue increase.
- Hide rate.- Number of ads hidden by user / number of show ads

Serving:

- Data Preparation Pipeline.
- Continual Learning Pipeline.
- Prediction Pipeline.