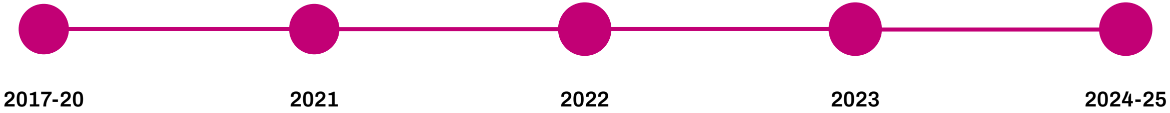
ROKT

Machine Learning in Industry

Ben Akres

My Journey



Studied Advanced Science and Arts at UNSW, majoring in Maths and Philosophy. Became interested in pursuing a career in ML at this time.

2017-20

Completed honours in Applied Mathematics with a focus on Data Mining and a thesis on clustering nodes in graphs.

2021

Joined Rokt as a Graduate ML Engineer. Worked on various model and feature changes to improve our CTR prediction accuracy.

2022

Worked on more complex projects, such as developing the ML systems for new product areas and unifying several of our models.

2023

Working as a Senior ML Engineer owning larger projects (e.g. improving our auction logic, developing new types of models, etc.).

2024-25

ROKT

About Rokt

Rokt connects global ecommerce in the moment that matters most

As the global leader in ecommerce technology Rokt acts as a trusted intermediary, helping companies seize the full potential of every transaction moment to grow revenue and acquire new customers at scale—from cart to confirmation page

6B+
Transactions will be powered by Rokt in 2025

$100M
Will be invested in R&D this year

40%+
YoY growth

400M+
Unique active users globally per annum

Unlock the moments that matter most through relevance in ecommerce, from cart to confirmation

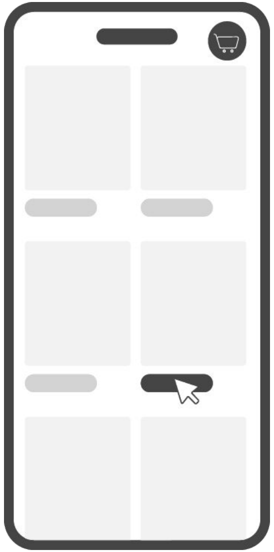
Leverage machine learning and customer relevance in the Transaction Moment to improve customer experiences and increase profit.

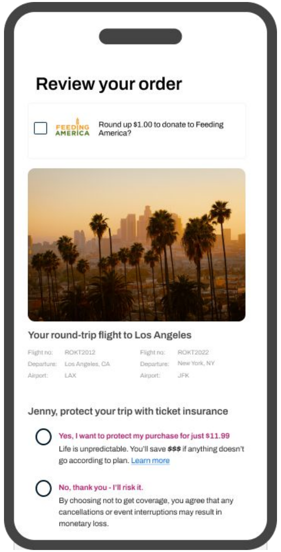
Demo

Transaction Moment

Product Selection | Cart | Payment | Confirmation

Product Centric Touchpoint | Customer Centric Touchpoints





Review your order

□ FEEDING AMERICA Round up $1.00 to donate to Feeding America?

Your round-trip flight to Los Angeles

Flight no: ROKT2012
Departure: Los Angeles, CA
Airport: LAX

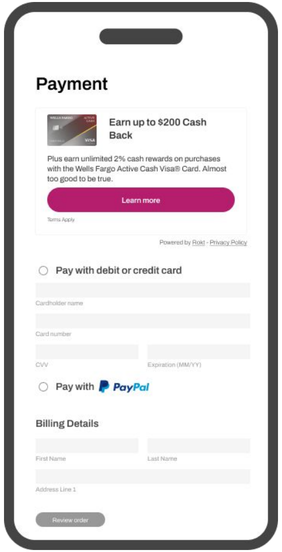
Flight no: ROKT2022
Departure: New York, NY
Airport: JFK

Jenny, protect your trip with ticket insurance

○ Yes, I want to protect my purchase for just $11.99
Life is unpredictable. You'll save ***$$$*** if anything doesn't go according to plan. Learn more

○ No, thank you - I'll risk it.
By choosing not to get coverage, you agree that any cancellations or event interruptions may result in monetary loss.

Upsell relevant products and services



Payment

Earn up to $200 Cash Back

Plus earn unlimited 2% cash rewards on purchases with the Wells Fargo Active Cash Visa® Card. Almost too good to be true.

Learn more

Terms Apply

Powered by Rokt - Privacy Policy

○ Pay with debit or credit card

Cardholder name

Card number

CVV | Expiration (MM/YY)

○ Pay with PayPal

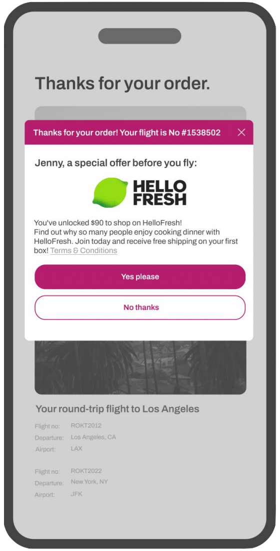
Billing Details

First Name | Last Name

Address Line 1

Review order

Transform a cost center into a profit center



Thanks for your order.

Thanks for your order! Your flight is No #1538502 ✕

Jenny, a special offer before you fly:

HELLO FRESH

You've unlocked $90 to shop on HelloFresh!
Find out why so many people enjoy cooking dinner with HelloFresh. Join today and receive free shipping on your first box! Terms & Conditions

Yes please

No thanks

Your round-trip flight to Los Angeles

Flight no: ROKT2012
Departure: Los Angeles, CA
Airport: LAX

Flight no: ROKT2022
Departure: New York, NY
Airport: JFK

Engage your customers with relevant offers

ROKT

Machine Learning at Rokt

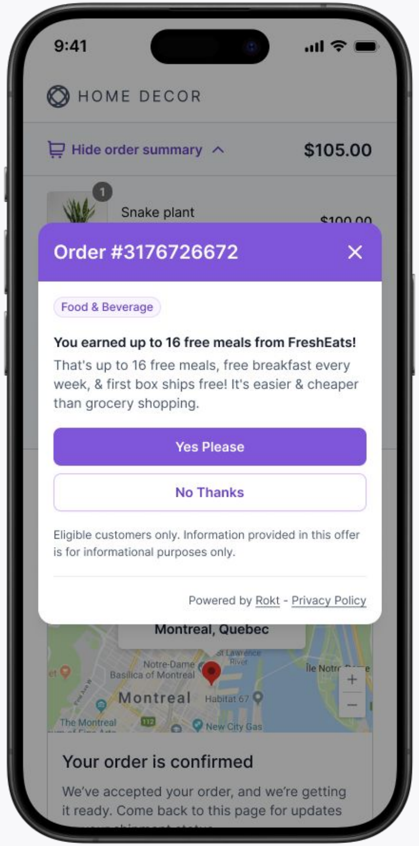
The Problem

Decide the most relevant experience to show a user (e.g. Upsell, Payment offer, 3rd party Ad, etc.), as well as how it is shown.

Deliver value by personalizing experiences and maximizing revenue across the entire Transaction Moment.

Models must:

- Perform at scale, processing millions of transactions per day with low latency.
- Be robust to changes in a dynamic environment.
- Deliver consistently on business objectives (e.g. revenue).



9:41

HOME DECOR

Hide order summary $105.00

Snake plant $100.00

Order #3176726672 ✕

Food & Beverage

You earned up to 16 free meals from FreshEats!

That's up to 16 free meals, free breakfast every week, & first box ships free! It's easier & cheaper than grocery shopping.

Yes Please

No Thanks

Eligible customers only. Information provided in this offer is for informational purposes only.

Powered by Rokt - Privacy Policy

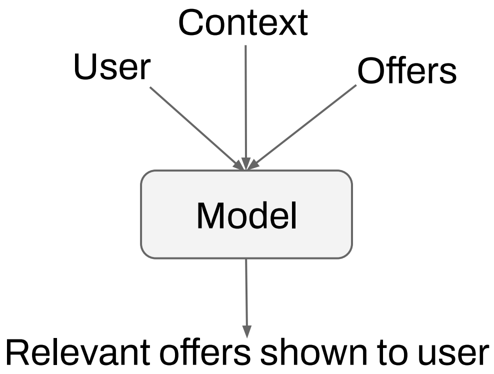
Montreal, Quebec

Your order is confirmed

We've accepted your order, and we're getting it ready. Come back to this page for updates

ML for Recommendations

- Deep Learning is state of the art for recommendation and CTR/CVR prediction tasks.
- Active area of research in both industry and academia with new ideas and model architectures being explored.
- Similar problem encountered in many large tech companies (Google, Facebook, Netflix, TikTok, etc.).



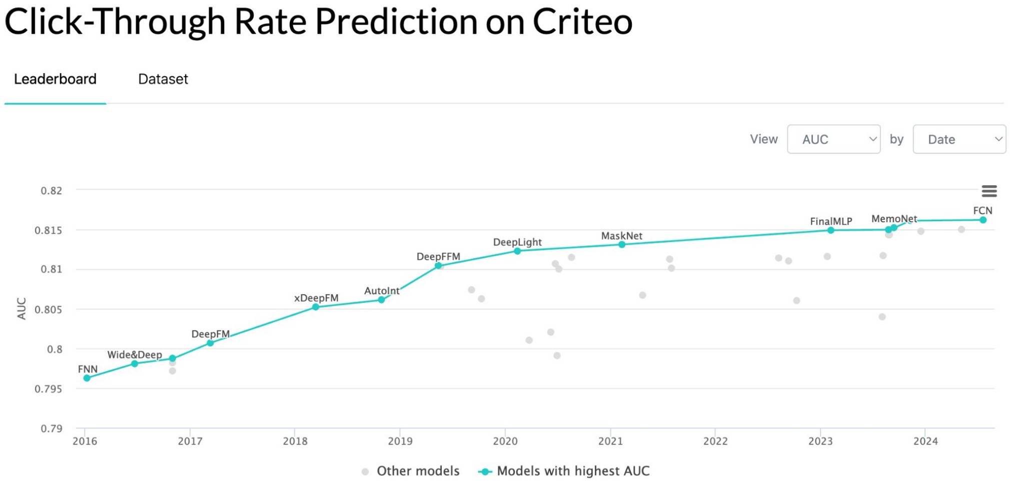
User

Context

Offers

Model

Relevant offers shown to user



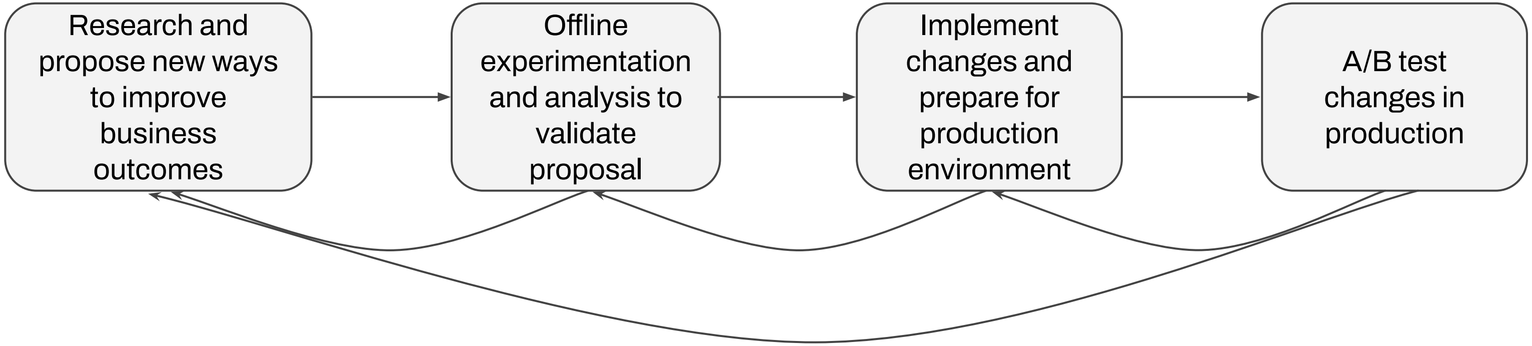
Click-Through Rate Prediction on Criteo

Leaderboard Dataset

View AUC by Date

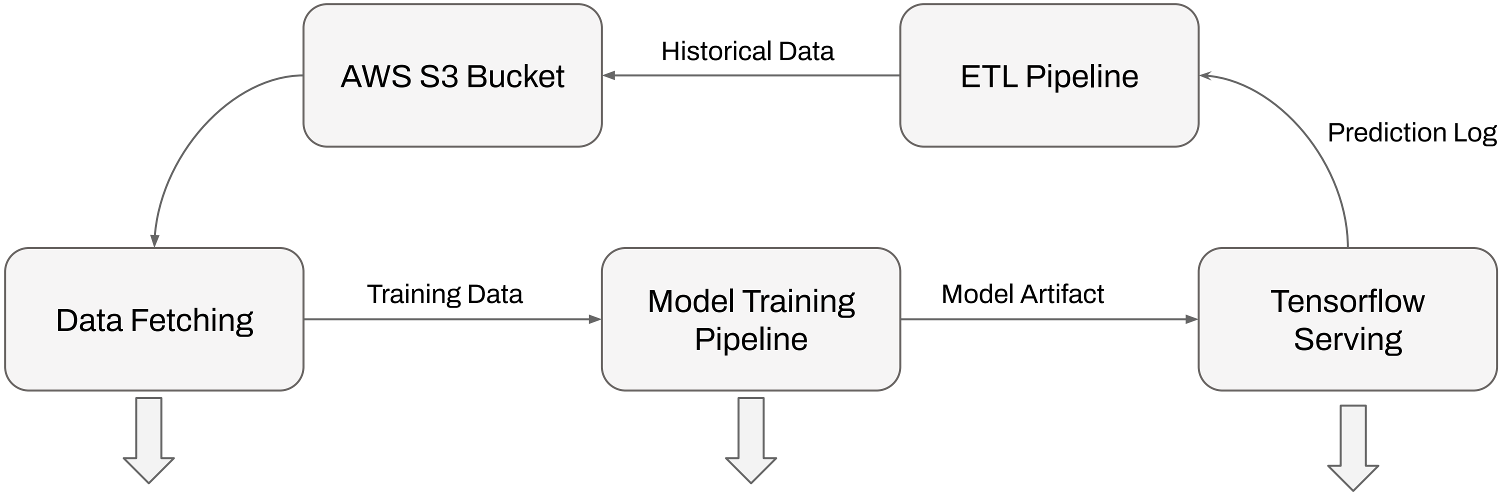
What Does an ML Engineer at Rokt Do?

- EDAs (exploratory data analysis), offline experiments and online experiments adding new feature(s) to the model(s).
- Research new architectures and algorithms to improve models.
- Perform deep dive investigations and analyses on how our models are performing and interacting with other systems throughout the business.
- Improve the way our model’s predictions are utilized by other systems.
- Propose and implement methods for optimizing different aspects of the product to improve business metrics.
- Ensure production ML systems are scalable, performant, reliable, low latency, robust, etc.


Research and propose new ways to improve business outcomes
Offline experimentation and analysis to validate proposal
Implement changes and prepare for production environment
A/B test changes in production

Taking Models to Production

ML Infrastructure in Production



AWS S3 Bucket

Historical Data

ETL Pipeline

Prediction Log

Data Fetching

Training Data

Model Training Pipeline

Model Artifact

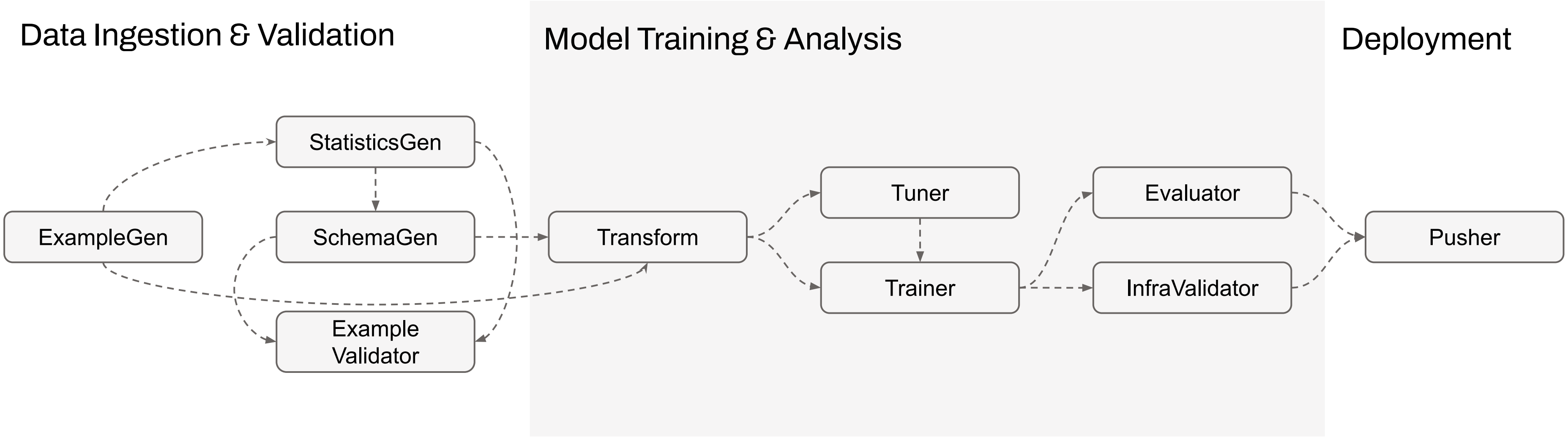
Tensorflow Serving

Fetches data from various sources (e.g. prediction logs, SQL tables, etc.)

Transforms data and uses it to train a model, evaluate its performance, and push it to production

A high performance serving system for ML models, designed for production environments

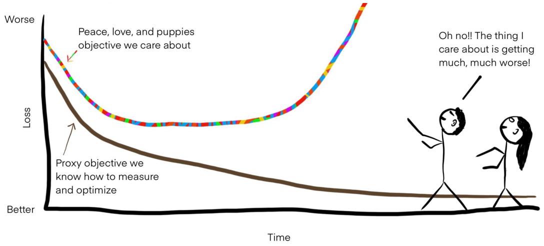
Model Pipeline Example



ROKT

Offline vs Online Performance

Source: https://sohl-dickstein.github.io/2022/11/06/strong-Goodhart.html



Worse
Peace, love, and puppies objective we care about
Oh no!! The thing I care about is getting much, much worse!
Loss
Proxy objective we know how to measure and optimize
Better
Time

- **Train/Serve skew**
  - Problem: The model may produce different outputs on training and serving data.
  - Example: A feature is missing during serving, resulting in inaccurate predictions.

- **Alignment with business objectives**
  - Problem: Metrics used to measure model performance during training and evaluation do not always reflect performance on business metrics.
  - Example: A video recommendation model that is used to show videos with the highest click rate may unintentionally promote clickbait, hurting long term engagement.
  - A model that performs well on model metrics may not perform well on business metrics!

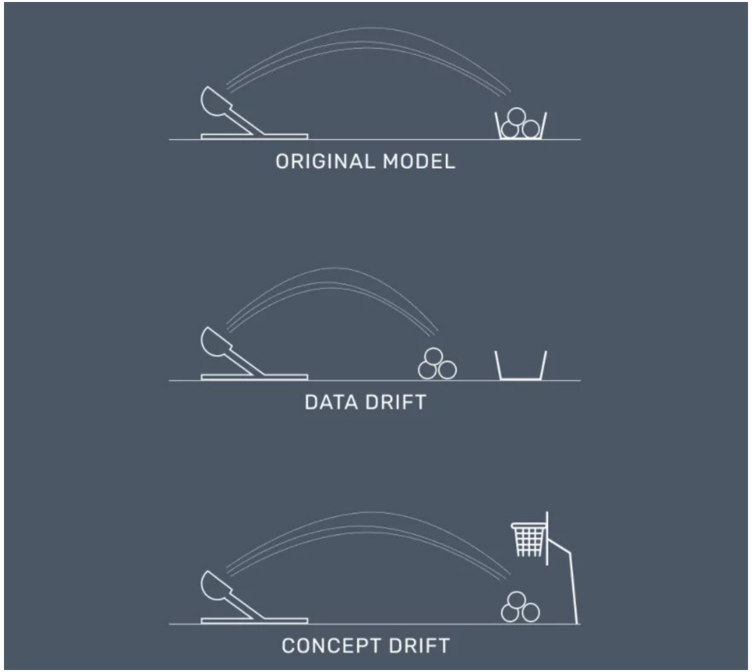
- **Solutions**
  - Ensure there are no discrepancies between data sources or feature engineering in online and offline environments (or ensure there is a single source of truth).
    - Observability
    - Skew detectors
  - Understand the business problem well enough to ensure the model is optimising for the right thing, and is being used in the right way.
  - A/B testing to ensure online model performance outperforms a benchmark or baseline.

Model Drift

- **Concept drift**
  - Problem: The input-output mapping that a model has learned may change over time.
  - Example: A model which accurately predicted housing prices 10 years ago will perform poorly today as the relationship between features and price has changed (due to a general increase in prices).

- **Data drift**
  - Problem: The distribution of input data can change over time.
  - Example: A model trained to recommend clothes based on data collected during the summer may perform poorly during winter as it has not trained on winter data.

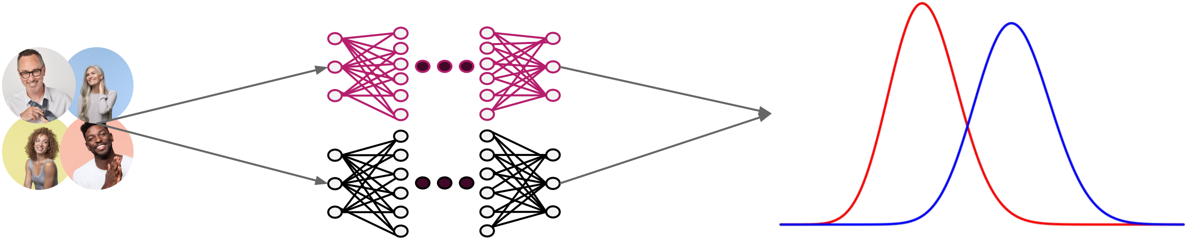
- **Solutions**
  - Observability
  - Scheduled retraining or online learning



Source: https://kdimensions.com/gallery-mlops/

ROKT

A/B Testing



- **All major model changes are A/B tested to ensure they improve business metrics.**

- **New model releases are done in stages to minimize disruptions.**
    1. (Optional) Allocate traffic to the model in shadow deployment.
    2. Allocate <5% of traffic to confirm the model behaves as expected.
    3. Allocate 50% of traffic to measure business metrics and real-time performance.
    4. Allocate 100% of traffic and celebrate!

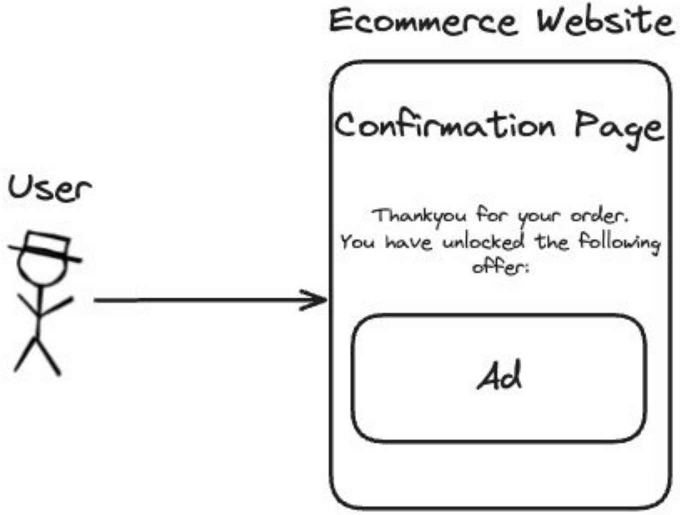
- **Challenges:**
    - How to split traffic (e.g. user based, session based, partner based, etc.)?
    - How long to run experiment for (ie. tradeoff between rapid iteration and confidence in results)?
    - How to measure statistical significance of results (e.g. different types of t-tests, Bayesian vs frequentist approach)?
    - How to analyse results (ie. changes over time, confounding variables, handling of outliers, Simpsons paradoxes, orthogonal experiments, etc.)?
    - How to measure second order effects (e.g. survivorship or selection biases, feedback loops, etc.)?

ROKT

ROKT

Recommender System Case Study

Ecommerce Recommender System Problem



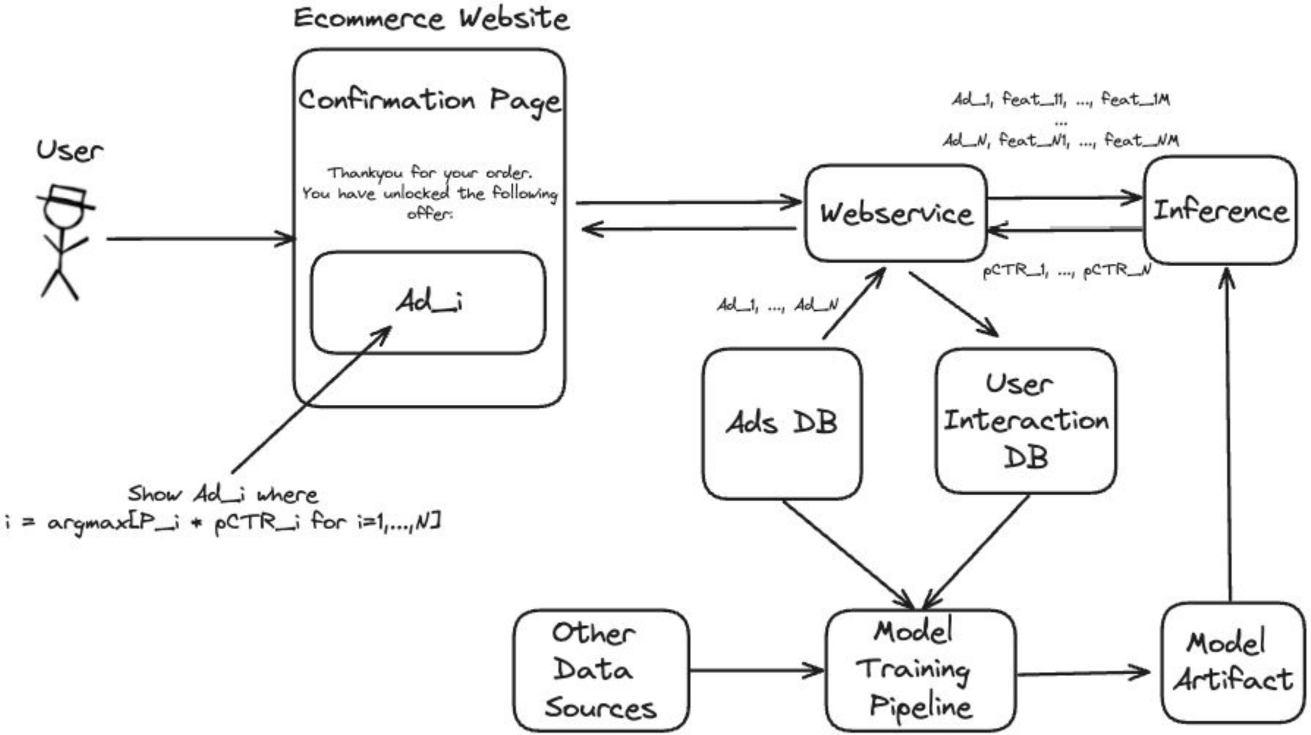
- Problem statement:
  - A user has just completed a transaction on an ecommerce website.
  - We can show them offers on the confirmation page to unlock additional revenue.
  - How can we decide which offer(s) to show?

- Ecommerce partner objective: maximise profit
- Advertiser objective: maximise return on Ad spend

- Requirements gathering:
  - Latency requirements
  - Expected throughput, volume of traffic, spikes in traffic
  - Number of offers available to select between
  - Data availability
  - Key business metrics

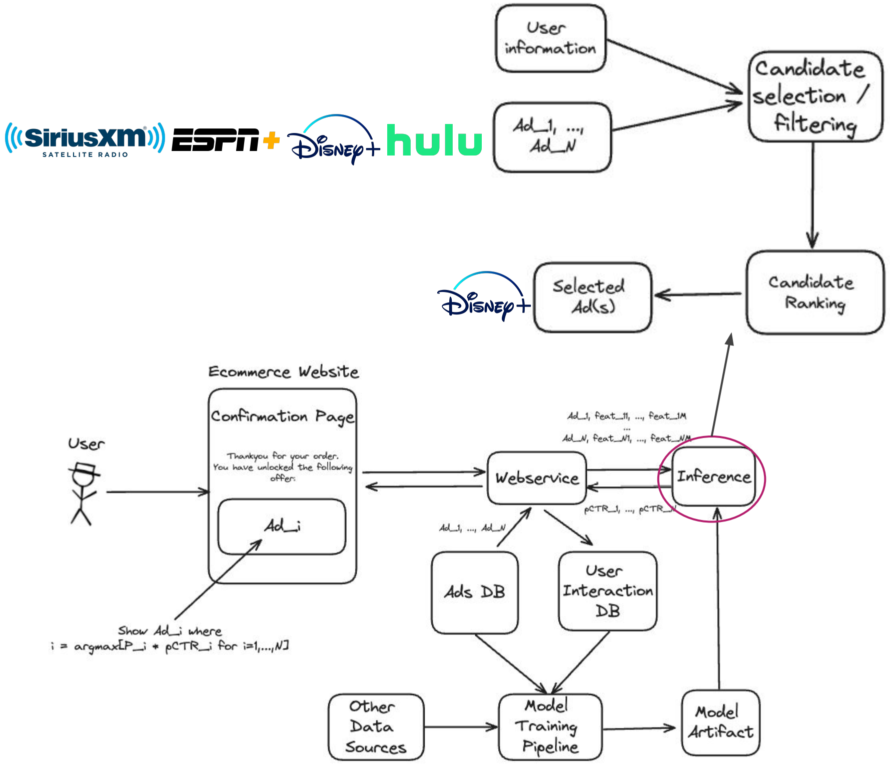
ROKT

Recommender System Design



ROKT

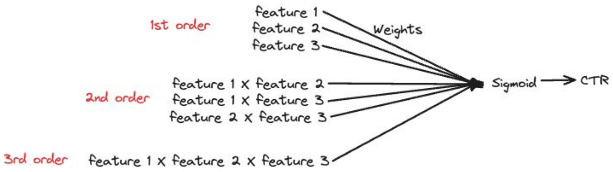
Two Stage Recommender System



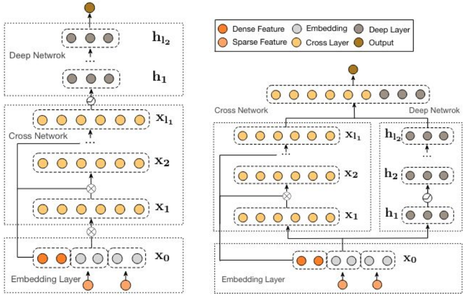
- If we have too many candidates it may be expensive or slow to rank them all - in these cases we may want to cut down the number of candidates
- Tradeoff between latency and accuracy - predicting for all candidates may be more accurate but this comes with a cost
- Filtering methods:
  - Simple low latency prediction model
  - Content filtering (e.g. show offers similar to items in the users cart)
  - Collaborative filtering (e.g. show offers that similar users have interacted with, or show offers similar to offers this user has interacted with in the past)
  - Approximate nearest neighbour lookup for user and/or offer embeddings
  - Business logic / targeting

Model Architectures for Recommender Systems

Deep Cross Networks (DCN)



Modelling feature interactions in logistic regression model with no cross layer

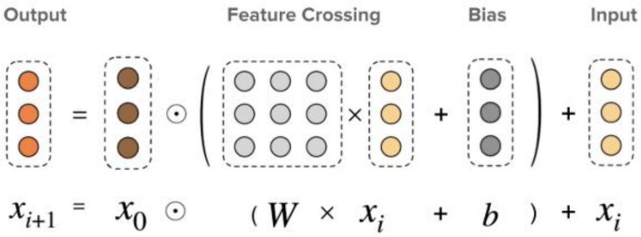


(a) Stacked

(b) Parallel

Figure 1: Visualization of DCN-V2. $\otimes$ **represents the cross operation in Eq. (1),** ***i.e.***, $x_{l+1} = x_0 \odot (W_l x_l + b_l) + x_l$.

Source: https://arxiv.org/pdf/2008.13535



$$x_{i+1} = x_0 \odot (W \times x_i + b) + x_i$$

Figure 2: Visualization of a cross layer.

- Feature interactions are important in recommender systems
  - For example, offer A performs better than B on average, but performs worse than B on Android devices
- You can add cross terms to your model explicitly through feature engineering, but identifying the right crosses can be time consuming
- Cross layers allow you to explicitly model feature interaction terms
- Stacking cross layers models higher order feature interactions
- A toy example using a DCN architecture can be found here: https://www.tensorflow.org/recommenders/examples/dcn
- DCN based architectures are state of the art on CTR prediction benchmarks

Model Architectures for Recommender Systems

Multigate Mixture of Experts (MMOE)

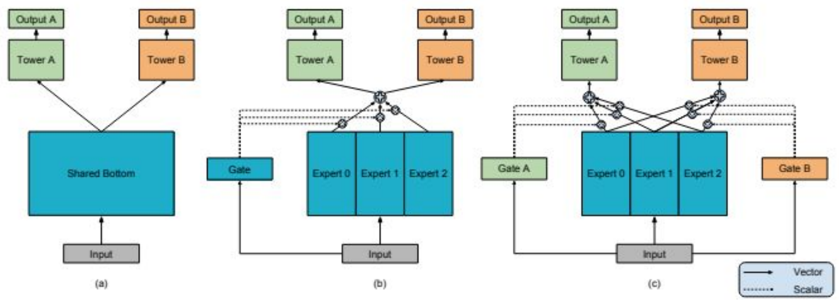


Figure 1: (a) Shared-Bottom model. (b) One-gate MoE model. (c) Multi-gate MoE model.

Source: https://dl.acm.org/doi/pdf/10.1145/3219819.3220007

- In some contexts we may require multiple predictions to be utilized by our recommender system (e.g. Click Rate, Conversion Rate, Cart Abandonment, user time on site, etc.).
- We can have different models for each of these tasks; however, if there are similarities between these tasks then using a multi-task model can lead to better results.
- In multi-task learning there can be tradeoffs between task specific relationships and inter-task relationships.
- “Each gating network can learn to ‘select’ a subset of experts to use conditioned on the input example. This is desirable for a flexible parameter sharing in the multi-task learning situation.”
- Additional considerations:
  - Weighting the loss functions from each task
  - Defining primary task(s) and auxiliary task(s)
  - Ensuring loss for each task converges at same time.

Model Architectures for Recommender Systems

Entire Space Multitask Modelling

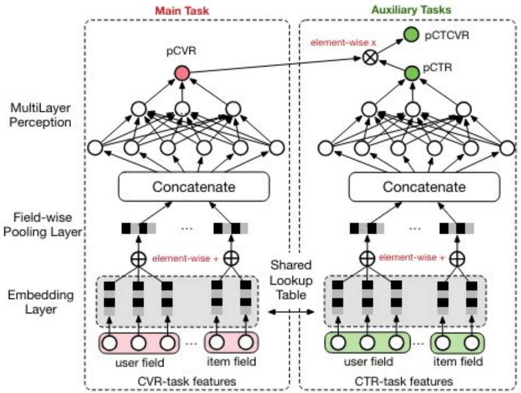


Figure 2: Architecture overview of ESMM for CVR modeling. In ESMM, two auxiliary tasks of CTR and CTCVR are introduced which: i) help to model CVR over entire input space, ii) provide feature representation transfer learning. ESMM mainly consists of two sub-networks: CVR network illustrated in the left part of this figure and CTR network in the right part. Embedding parameters of CTR and CVR network are shared. CTCVR takes the product of outputs from CTR and CVR network as the output.

Source: https://dl.acm.org/doi/pdf/10.1145/3219819.3220007

$$\underbrace{p(y=1,z=1|\mathbf{x})}_{pCTCVR} = \underbrace{p(y=1|\mathbf{x})}_{pCTR} \times \underbrace{p(z=1|y=1,\mathbf{x})}_{pCVR}$$

$$L(\theta_{cvr},\theta_{ctr}) = \sum_{i=1}^{N} l(y_i, f(\mathbf{x}_i;\theta_{ctr})) + \sum_{i=1}^{N} l(y_i\&z_i, f(\mathbf{x}_i;\theta_{ctr})\times f(\mathbf{x}_i;\theta_{cvr}))$$

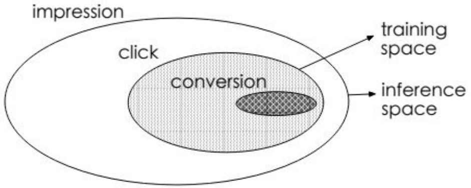
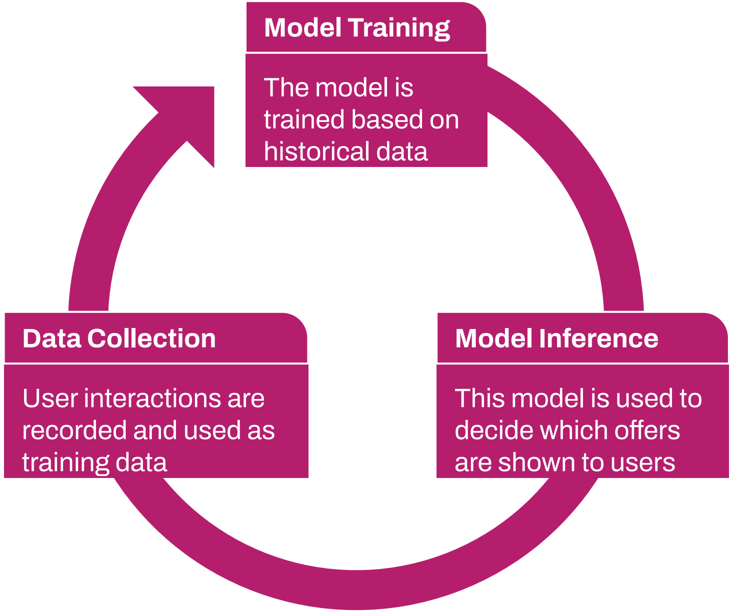


Figure 1: Illustration of sample selection bias problem in conventional CVR modeling. Training space is composed of samples with clicked impressions. It is only part of the inference space which is composed of all impressions.

- In recommender systems there are known relationships between the quantities we would like to predict - we can use this information when designing our model architecture
- Modelling CVR can be hard due to sparse labels and selection bias in the training set
- ESMM helps these problems by sharing feature embeddings with the click prediction task (helping with label sparsity) and allowing us to train on samples from the whole space of impressions (helping with selection bias)

Additional Challenges

Exploration vs Exploitation



Model Training

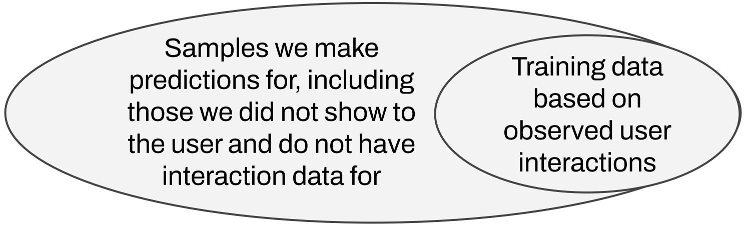
The model is trained based on historical data

Model Inference

This model is used to decide which offers are shown to users

Data Collection

User interactions are recorded and used as training data



Samples we make predictions for, including those we did not show to the user and do not have interaction data for

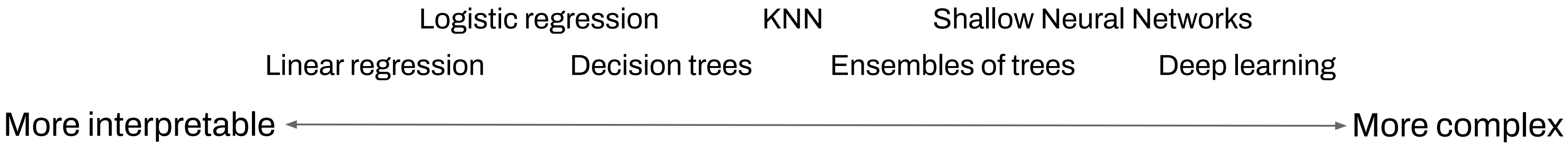
Training data based on observed user interactions

Recommender systems create feedback loops; the choice of recommendation determines the data available for training.

Multi-Armed Bandit Problem

- If we always show what we think is the best offer (exploiting), then we will not have the opportunity to learn about potentially better offers (exploring).
- Need to strike the right balance between exploration and exploitation.
- Epsilon-Greedy is an example of a simple strategy to handle this (explore a small percent of the time, exploit otherwise).

Model Interpretability



Logistic regression

KNN

Shallow Neural Networks

Linear regression

Decision trees

Ensembles of trees

Deep learning

More interpretable

More complex

More complex models can be less interpretable.

An advantage of tree based models is that they are easy to interpret.

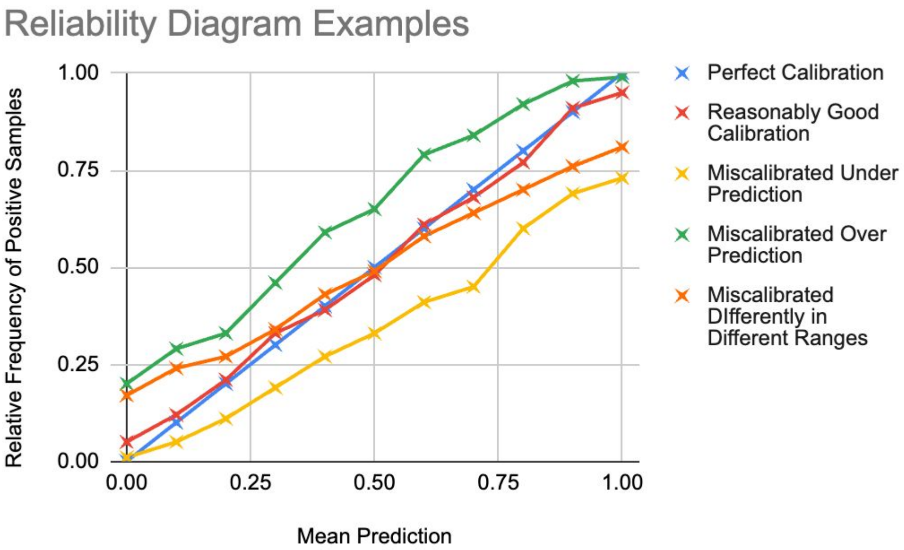
- It is easy to understand how decisions are made for an individual decision tree.
- When considering ensembles of trees we can gauge feature importance from factors like the information gain, or number of splits on a feature.

Deep learning models are hard to interpret.

- There are model agnostic methods for obtaining feature importance, such as feature permutation and SHAP, but these are often computationally expensive.
- Ongoing area of research.

Model Calibration

- In some contexts it is not sufficient to have models which correctly identify the best offer to show; we may require our model predictions to be well calibrated probabilities.
- A classifier is considered well calibrated if a prediction of *x* indicates that the probability of the sample having a positive label is actually *x*.
- Some classification metrics (like AUC) ignore model calibration.
- Deep learning models can be overconfident in their predictions, particularly on unbalanced datasets (upsampling or sample weighting can exacerbate this).



Questions

ROKT