COVEO Final Presentation

Teammates:

Tina Yang Zhou
Ahana Ahluwalia
Martin Dallaire
Prakash Gawas

Introduction

Background : For e-commerce platforms, it is important to understand whether a user is a “buyer” or “window shopper” based on clickstream data.

Goal: to use in-session data to predict whether the customer carry out purchase within the session, and eventually, to predict the sales revenue.

What has been done: product recommendation models in COVEO competitions.

Task: Predict conversion rate based on set amount of time

Given X amount of time for a session, how likely is an user to buy something?

Things to note:

~50% of buying sessions had at least 20 non-buying events before a “buy” event.

~50% of all sessions only had 3 or fewer events in total.

~50% of buyers spent at least 12.7 minutes before buying.

Given the last point, we chose **10 minutes** as our observation period.

Model: Gradient-boosted Decision Trees (using LightGBM)

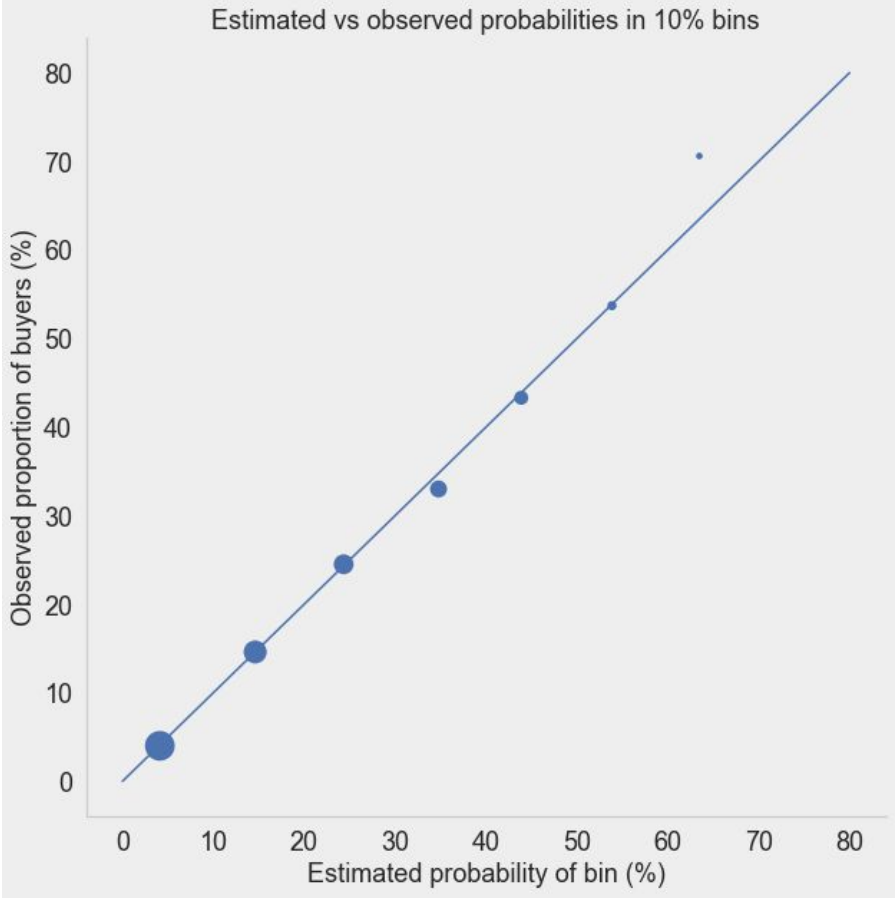
Using LightGBM, we achieve a very modest (but consistent) improvement over a zero-baseline in the task of predicting whether people will buy after observing the first 10 minutes of a session.

Zero-baseline: **88.1%** of user session (with at least one add-to-cart) do not make a purchase after the 10 minutes mark (purchases before the 10 minutes mark are not counted)

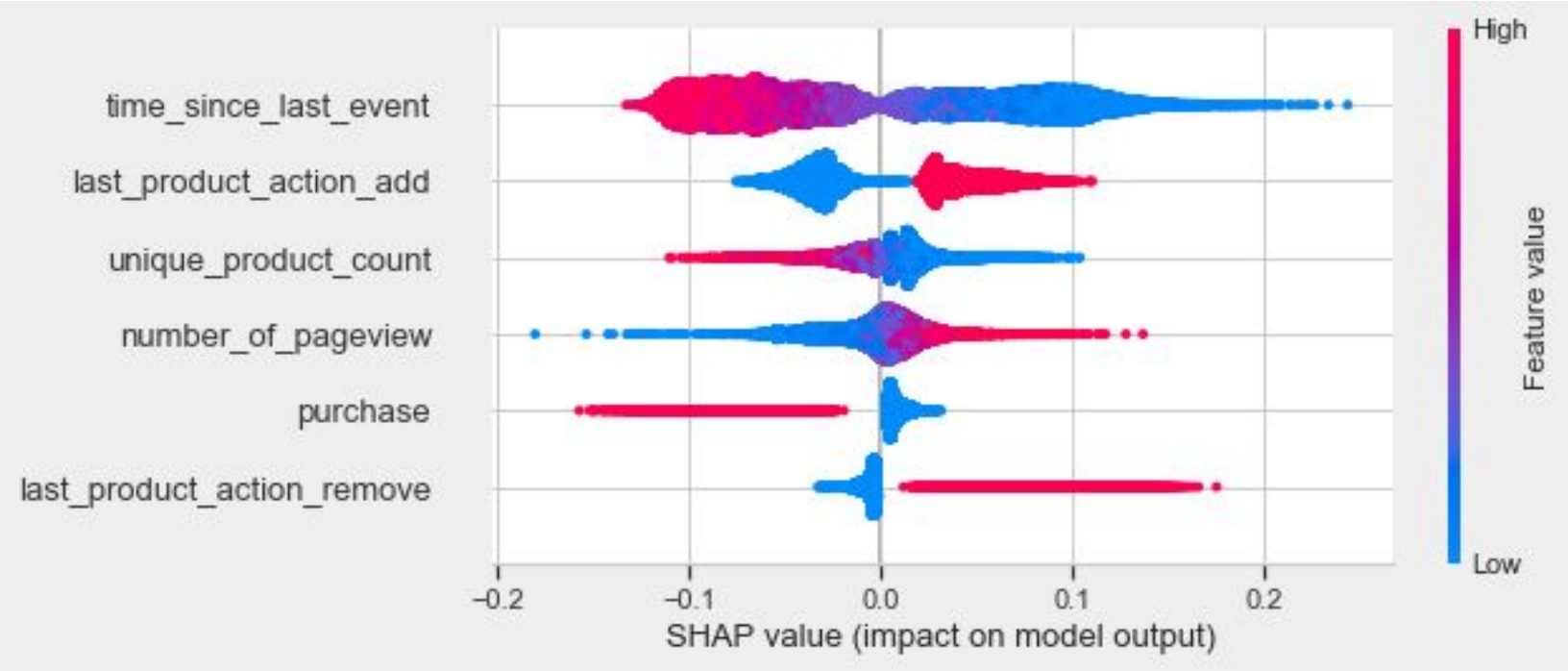
Prediction model: **88.2%** accuracy, which seems disappointing, but...

Output probabilities seem heavily reliable

- Our model's outputted probabilities match the observed outcomes.
- The lackluster accuracy mentioned previously is because very, very few sessions have a higher than 50% chance of converting (according to our model).
- Could be useful to target likely buyers.



Top features used by the model



Average user versus likely user

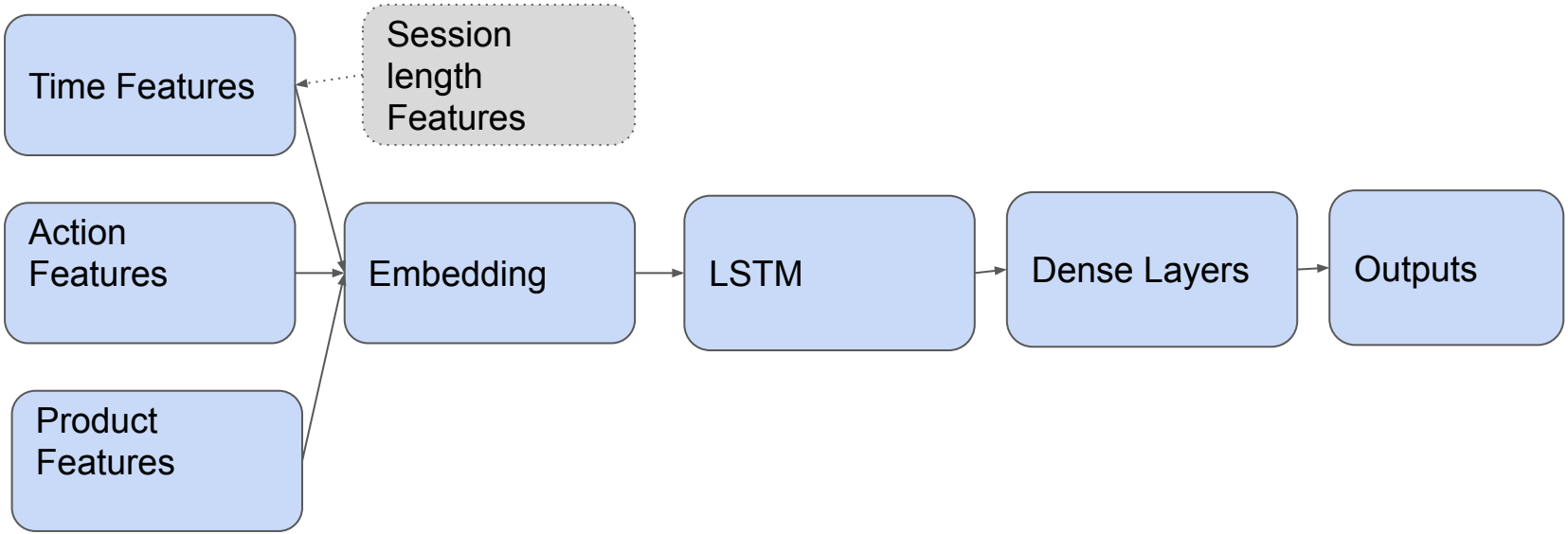
Conclusions

- We can use a fast model to decide which sessions are likely to convert into purchases.
- The feature importance of the model gives insight on “good” session behaviors.
- The model can easily be modified to take into account more historical user-based session information.
- While we weren’t supposed to know which sessions were from recurring users, the model was able to figure out some of them by looking at cart removal, and those were our most likely buyers.
- In the future, we should try smaller amounts of time.

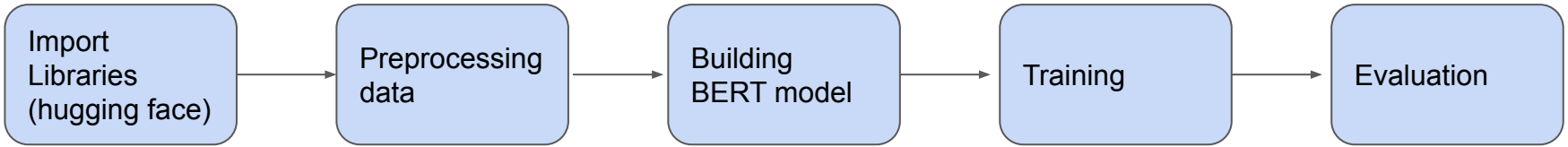
Future Work

- Deep learning-based model can be applied to analyze sequential pattern data
- Models to explore: LSTM
  BERT
  Similarity based KNNs

LSTM (long short term memory) model



BERT pretrained model



Similarity based Knn prediction

Motivation:

From https://arxiv.org/pdf/2011.03424.pdf

session-based recommendation algorithms and trivial extensions thereof. Our comparison, to some surprise, revealed that *(i)* simple techniques based on nearest neighbors consistently outperform recent neural techniques and that *(ii)* session-aware models were mostly not better than approaches that do not use long-term preference information. Our work therefore not


```
[{
    "query": [{
        "session_id_hash": "00000114e1075962f022114fcfc17f2d874e694ac5d2010985bbba0a595340db",
        "query_vector": null,
        "clicked_skus_hash": null,
        "product_skus_hash": null,
        "server_timestamp_epoch_ms": 1552423391039,
        "event_type": "event_product",
        "product_action": "detail",
        "product_sku_hash": "cf2f88cb43c1713538f7dfd2aa498a2cb9ebc0c99feeac91820b9d9fdfe981f6",
        "hashed_url": "0aa1084eddfb08e4dffbb5a2aa98a5e9679382d982dd9707c1bc897cfdeafbd1",
        "is_search": false
    }, {
        "session_id_hash": "00000114e1075962f022114fcfc17f2d874e694ac5d2010985bbba0a595340db",
        "query_vector": null,
        "clicked_skus_hash": null,
        "product_skus_hash": null,
        "server_timestamp_epoch_ms": 1552424389158,
        "event_type": "event_product",
        "product_action": "add",
        "product_sku_hash": "cf2f88cb43c1713538f7dfd2aa498a2cb9ebc0c99feeac91820b9d9fdfe981f6",
        "hashed_url": "83b4fdad686c1be4eba335f70d23ae202b84b6153e109edb6279901e1bea5120",
        "is_search": false
    }, {
        "session_id_hash": "00000114e1075962f022114fcfc17f2d874e694ac5d2010985bbba0a595340db",
        "query_vector": null,
        "clicked_skus_hash": null,
        "product_skus_hash": null,
        "server_timestamp_epoch_ms": 1552424698656,
        "event_type": "pageview",
        "product_action": null,
        "product_sku_hash": null,
        "hashed_url": "433b0e71df1fe9a8d1f45647545701f6108414c40eef76792a5a7404cf9a3067",
        "is_search": false
    }],
    "prediction": ["4945f2fa8e87cb7501702ed3dce26253296eae7a8f670fd5a98ecc3d10b1c40a", "6ff8d0f30bbe66cfec7d87fc7e22bd8b1defd47ff4aaaa4cd13aaa6c46003a18"]
}, {
```


select only ‘detail’ events

create sequences of products

encode them using binary vectors over the itemspace

calculate cosine similarity between the binary vectors

$$score_{\text{SKNN}}(i, s) = \Sigma_{n \in N_s} sim(s, n) \cdot 1_n(i)$$

$$score_{\text{S-SKNN}}(i, s) = \Sigma_{n \in N_s} sim(s, n) \cdot w_n(s) \cdot 1_n(i)$$

Similarity based Knn prediction

```
[[963806208265782664693181337441840038390310815909972433966269826990945381229 08,
  442970120952678747817919545863618900434514616010028856944722484893909401143 71,
  963806208265782664693181337441840038390310815909972433966269826990945381229 08,
  963806208265782664693181337441840038390310815909972433966269826990945381229 08,
  432600027284823405601601647114820070213975935948393235090159295447914510255 59,
  450994888083659219000557694301299543443501666631079583026486217216904482055 20],
 [575959484655060795905049287011352932517397420647361194937123155640684507727 31,
  405716685466554057494485778807795728261106008109684507062258093157979367534 96,
  575959484655060795905049287011352932517397420647361194937123155640684507727 31,
  575959484655060795905049287011352932517397420647361194937123155640684507727 31],
```

cosine similarity rank

```
4037: array([0.8660254]),
917: array([0.70710678]),
2074: array([0.70710678]),
2617: array([0.70710678]),
2776: array([0.70710678]),
3383: array([0.70710678]),
1800: array([0.63245553]),
1647: array([0.61237244]),
2467: array([0.61237244]),
2831: array([0.61237244]),
3625: array([0.61237244]),
155: array([0.57735027]),
784: array([0.57735027]),
895: array([0.57735027]),
1801: array([0.57735027]),
2095: array([0.57735027]),
```

Max $\underset{(x\_s)_{s \text{ in } S}}{\text{Sum}_{s \text{ in } S}} [\, x\_s \cdot \text{Expected}[\text{purchase\_s}] \,]$

$(y_i^L, y_i^U, y_i)_{i \text{ in } I}$

Subject to:

$z_i^U <= y_i^U \cdot f_i^U$ for every i in I

$z_i^L <= y_i^L \cdot f_i^L$ for every i in I

$y_i^L + y_i^U - 1 <= y_i$ for every i in I

$\Pi_{i \text{ in } I}\, y_i >= x_s$ for every s in S

$\text{Sum}_{s \text{ in } S}[\, x_s \,] <= 1$

x_s in [0,1]

y_i^L, y_i^U, y_i in {0,1}

Motivation and avenues for future work

We have studied the dataset from the following two perspectives:

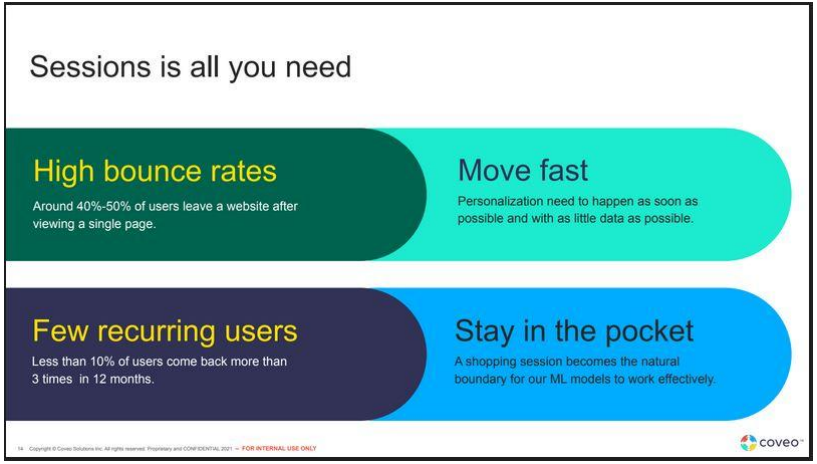
1. Purchase prediction
2. Segmentation

We want to use these information to guide the decision making of COVEO.

Problem of COVEO:

The personalization of the session should be done quickly.

We address this problem from a methodological point of view.



Thank you.