milestone_2_00951537

June 15, 2024

1 Preamble

1.0.1 Data Download - Preferred, fully replicable to the below

> **Note**: Due to submission size limitations, the datasets were **not** included in the repost submission. Instead, I have made them publicly downloadable from S3.

The preferred method to retrieve the data is to run `dvc pull` in the CLI. This will ensure that the exact same samples are retrieved to the ones used in the report. See documentation for DVC for reference. This depends on installing `dvc` and `dvc-s3` in the environment.

1.0.2 Data Download - Alternative

Run the following line once in order to retrieve the data samples from AWS S3. Below, we pass 100000 as the sample argument. Before doing so, please ensure that `boto3` and `s3fs` are installed. Refer to `requirements.txt` for further dependencies. `{python} %run scripts/s3_data_retrieval/retrieve_samples_from_s3.py 100000`

2 Introduction

The global digital advertising market is worth approximately $602 billion today. Due to the increasing rate of of online participation since the COVID-19 pandemic, this number has been rapidly increasing and is expected to reach $871 billion by the end of 2027 (eMarketer, 2024). Many of the of the major Ad platforms such as Google, Facebook and Amazon operate on a cost-per-user-engagement pricing model, which usually means that advertisers get charged for every time a user clicks on an advertisment. This means that these platforms are incentivized to make sure that the content shown to each user is as relevent as possible in order to maximize the number of clicks in the long term. Attaining accurate Click-Through Rate (CTR) prediction is a necessary first step for Ad persionalization, which is why study of CTR prediction methods have been an extremely active part of Machine Learning research over the past through years.

Initially, shallow prediction methods such as XGBoost (Cite), Factorization Machines (Cite) and Field-Aware Factorization Machines (Cite) have been used for CTR prediction. However, these methods have often been shown to be unable to capture the higher order feature interactions in the sparse multy value categorical Ad Marketplace datasets (Cite). Since then, Deep Learning methods have been shown to show superior predictive ability on these datasets. The focus of my reasearch project is therefore to explore the merits of different Deep Learning architechtures for click-through rate prediction

In the following report, I explore the relevant datasets and simulations that I will be using throughout my research project. In the first section, I perform an exploratory data analysis on three widely adopted benchmark CTR prediction datasets; the KDD12 (Aden, 2012), Avazu (Wang and Cukierski, 2014) and Criteo (Tien et al, 2014) datasets. In the second section, I then explore possible ways of simulating the ad marketplace environment in order to test the reinforcement learning framework.

3 Data Analysis and Pre-processing

I begin below by first introducting the three datasets widely used as benchmarks in CTR prediction research.

3.0.1 KDD12

The **KDD12** dataset was first released for the KDD Cup 2012 competition (Cite), with the original task being to predict the number of clicks for a given number of impressions. Each line represents a training instance derived from the session logs for the advertizing marketplace. In the context of this dataset, a "session" refers to an interaction between a user and the search engine, containing the following components; the user, a list of adverts returned by the search engine and shown (impressed) to the user and zero or more adverts clicked on by the user. Each line in the training set includes:

- **Click and Impression counts**: The click counts were the original target variable when the dataset was first released for the competition. As done in (Cite Song and Others), this dataset can be adapted to CTR prediction by simply calculating the CTR for each instance by dividing the Click counts by the Impression counts.
- **Session features**: These include *session depth* (the number of ads impressed in a session) as well as the tokenized query phrase that the user entered into the search engine.
- **User features**: Encoded gender and age group for the user, if known.
- **Ad features**: Display URL, ad ID, advertiser ID and encoded title, description and purchased key words.

```
Snapshot of KDD12 training data:

   Click  Impression            DisplayURL      AdID  AdvertiserID  Depth  \
0      0           1  12057878999086460853  20157098         27961      1
1      0           1  12057878999086460853  20221208         27961      2
2      0           1  12057878999086460853  20183701         27961      1
3      0           1  12057878999086460853  20183690         27961      1
4      0           1   3029113635936639912  10397010         24973      2

   Position   QueryID KeywordID TitleID DescriptionID   UserID
0         1     75606     15055   12391         13532  1350148
1         1      2977      1278    3054          4561  1350148
2         1  18594855       227     543           642  1350148
3         1   4260473     34048  175983        155050  1350148
4         2      2977      1274    2570         26091  1350148
```

3.0.2 Avazu

The **Avazu** dataset was originally released in 2014 for a CTR prediction Competition on Kaggle (Cite Avazu). The data is composed of 11 days worth mobile ad marketplace data. Much like the KDD12 dataset above, this dataset contains features ranging from user activity (clicks), user identification (device type, IP) to ad features. Notible differences to the KDD12 dataset include the fact that Avazu contains an “hour” feature (enabling the establishment of sequentiality of behaviours) and the fact that Avazu does not seem to contain query and ad texts.

```
Snapshot of Avazu training data:

                     id  click      hour    c1 banner_pos   site_id  \
0  15674134821169810910      1  14102300  1005          0  85f751fd
1  15674278914362889244      0  14102300  1005          0  85f751fd
2   1567455966106046075      0  14102300  1005          0  26fa1946
3  15674616734887926359      0  14102300  1005          0  85f751fd
4  15674670592044781339      0  14102300  1005          0  85f751fd

  site_domain site_category    app_id app_domain  … device_type  \
0    c4e18dd6      50e219e0  e71aba61   2347f47a  …           1
1    c4e18dd6      50e219e0  6f8bcb0f   2347f47a  …           1
2    e2a5dc06      3e814130  ecad2386   7801e8d9  …           1
3    c4e18dd6      50e219e0  53de0284   d9b5648e  …           1
4    c4e18dd6      50e219e0  a0fc55e5   2347f47a  …           1

  device_conn_type    c14  c15  c16   c17  c18  c19     c20  c21
0                0  21676  320   50  2495    2  167      -1   23
1                0  20476  320   50  2348    3  427  100005   61
2                0  20362  320   50  2333    0   39      -1  157
3                0  21611  320   50  2480    3  297  100111   61
4                0  20361  300  250  2333    0   39      -1  157

[5 rows x 24 columns]
```

3.0.3 Criteo

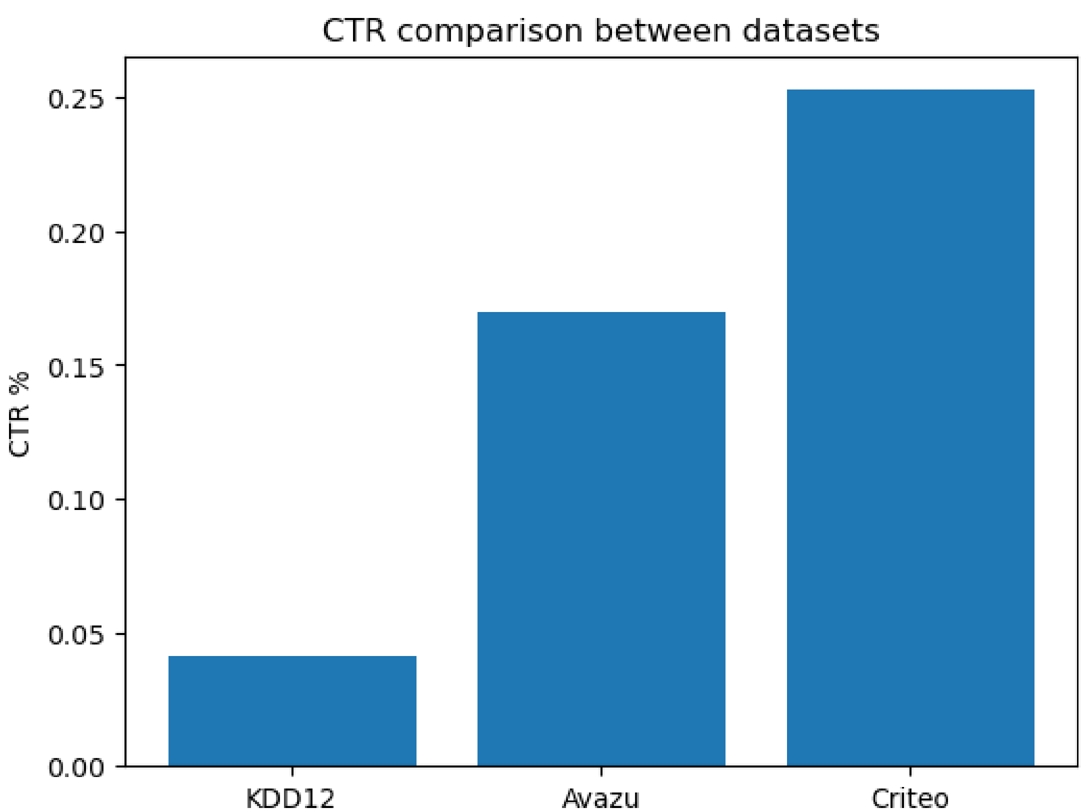
Finally, the Criteo dataset is another benchmark CTR prediction dataset that was originally released on Kaggle for a CTR prediction compitition. The original dataset is made up of 45 Million user’s click activity, and contains the click/no-click target along with 26 categorical feature fields and 13 numerical feature fields. Unlike the other two datasets however, the semintic significance of these fields is not given - they are simply labelled as “Categorical 1-26” and “Numerical 1-13” respectively.

```
Snapshot of Criteo training data:

   click  int_1  int_2  int_3  int_4    int_5  int_6  int_7  int_8  int_9  \
0      0    NaN      1    2.0    5.0  27586.0   32.0    2.0   14.0   21.0
1      1   14.0      1    1.0    8.0    276.0   14.0   41.0    9.0   10.0
2      0    NaN      1   27.0   25.0      NaN    NaN    0.0   54.0   55.0
```

```
3      0    0.0    442    1.0    1.0   3029.0   58.0    2.0   13.0   44.0
4      0    0.0     -1    2.0    1.0   1167.0   88.0   23.0   19.0  673.0

   …    cat_17    cat_18    cat_19    cat_20    cat_21 cat_22    cat_23  \
0  …  07c540c4  bdc06043       NaN       NaN  6dfd157c    NaN  32c7478e
1  …  e5ba7672  87c6f83c       NaN       NaN  0429f84b    NaN  be7c41b4
2  …  2005abd1  87c6f83c       NaN       NaN  15fce809    NaN  be7c41b4
3  …  d4bb7bd8  cdfa8259       NaN       NaN  20062612    NaN  dbb486d7
4  …  27c07bd6  5bb2ec8e  49b8041f  b1252a9d  bff87997    NaN  32c7478e

     cat_24    cat_25    cat_26
0  ef089725       NaN       NaN
1  c0d61a5c       NaN       NaN
2  f96a556f       NaN       NaN
3  1b256e61       NaN       NaN
4  3fdb382b  f0f449dd  49d68486

[5 rows x 40 columns]
```

3.0.4 Target Variable Analysis

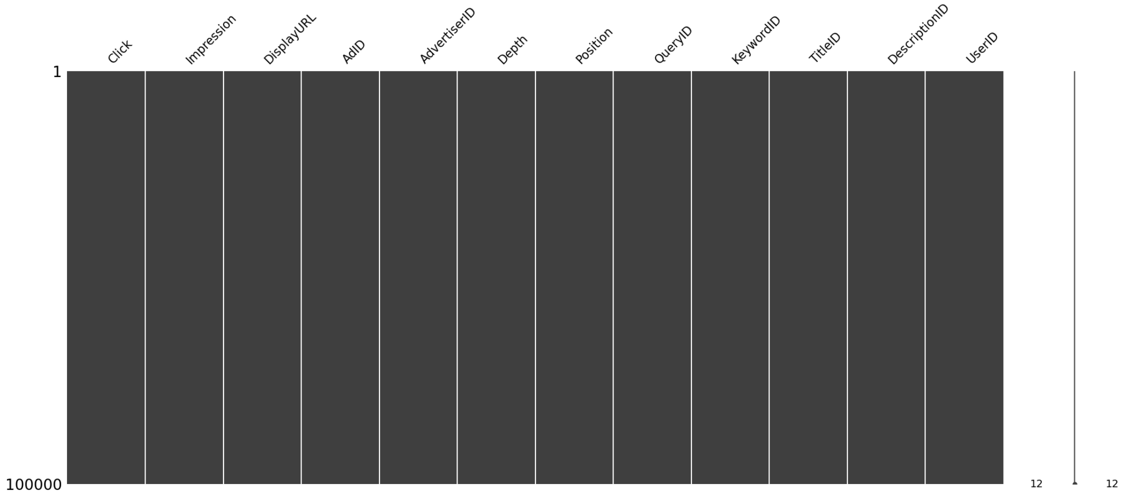
The figure below shows that the three datasets have vastly different average Click Through Rates per instance. The average CTR for the KDD12 dataset is only 3.4%, whereas the Criteo dataset is 25.6%.

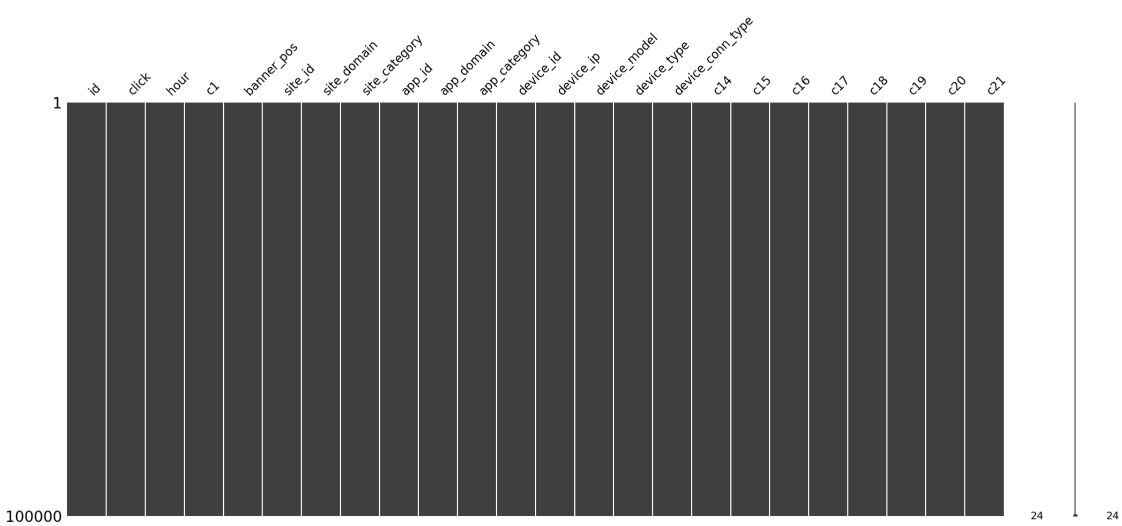


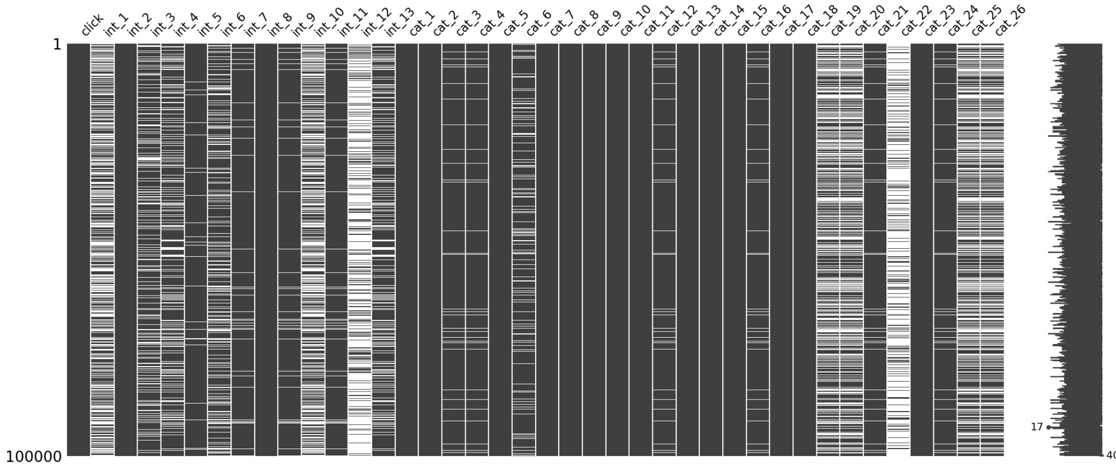
3.0.5 Missingness

Below, I take a look at whether or not our dataset has any missing values.

```
Missingness matrix for KDD12 dataset:

<Axes: >

Missingness matrix for Avazu dataset:

<Axes: >

Missingness matrix for Criteo dataset:

<Axes: >
```







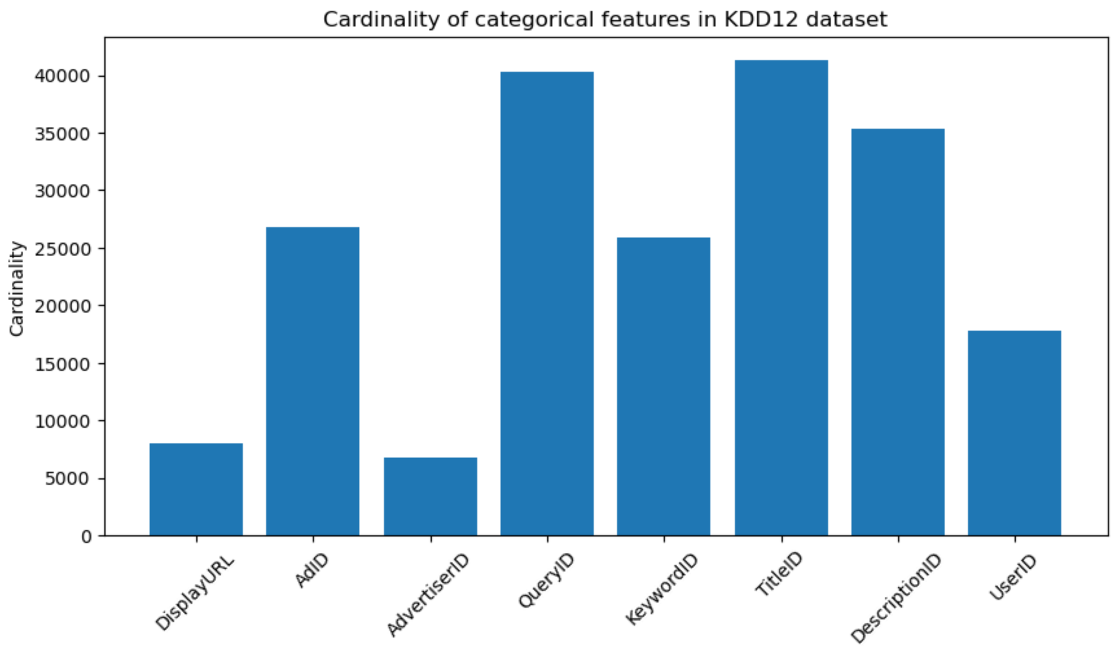
Above we see that both KDD12 and Avazu tend to be well populated. However, we also see that Criteo has some missing values. Below I proceed by imputing the missing values using Sklearn's KNN Imputer

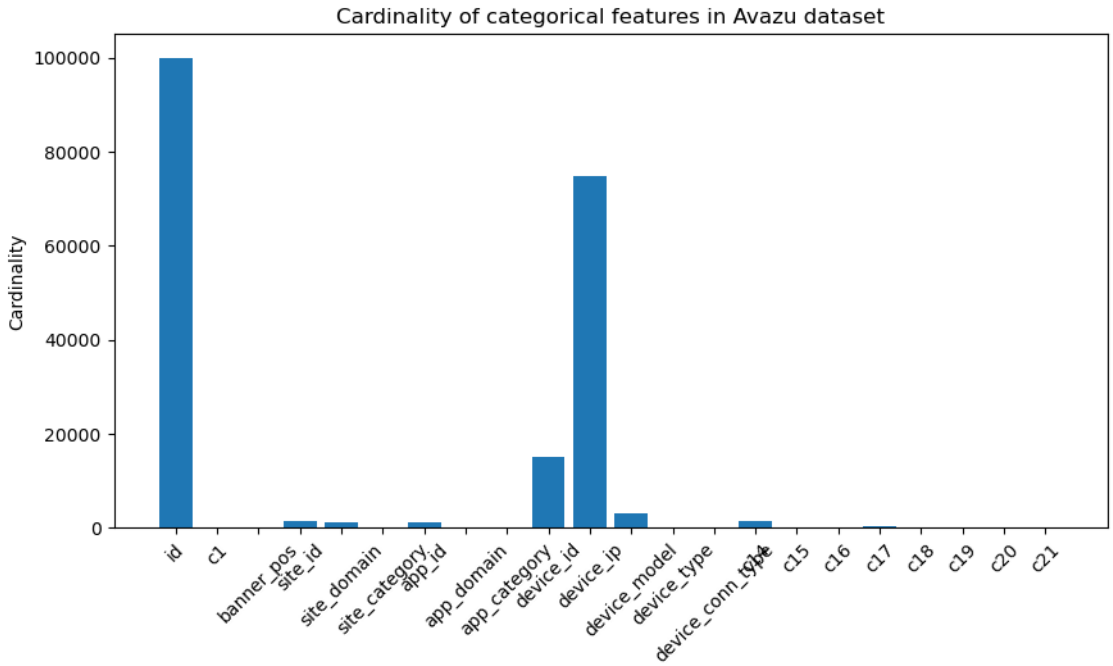
```
/opt/conda/lib/python3.10/site-packages/sklearn/impute/_iterative.py:801:
ConvergenceWarning: [IterativeImputer] Early stopping criterion not reached.
  warnings.warn(

   click  int_1  int_2  int_3  int_4   int_5  int_6  int_7  int_8  int_9  …  \
0      0      0      1      2      5   27586     32      2     14     21  …
1      1     14      1      1      8     276     14     41      9     10  …
2      0      0      1     27     25  251808    840      0     54     55  …
3      0      0    442      1      1    3029     58      2     13     44  …
4      0      0     -1      2      1    1167     88     23     19    673  …

   cat_6_missing  cat_12_missing  cat_16_missing  cat_19_missing  \
0              0               0               0               1
1              0               0               0               1
2              0               0               0               1
3              0               0               0               1
4              0               0               0               0

   cat_20_missing  cat_21_missing  cat_22_missing  cat_24_missing  \
0               1               0               1               0
1               1               0               1               0
2               1               0               1               0
3               1               0               1               0
4               0               0               1               0
```

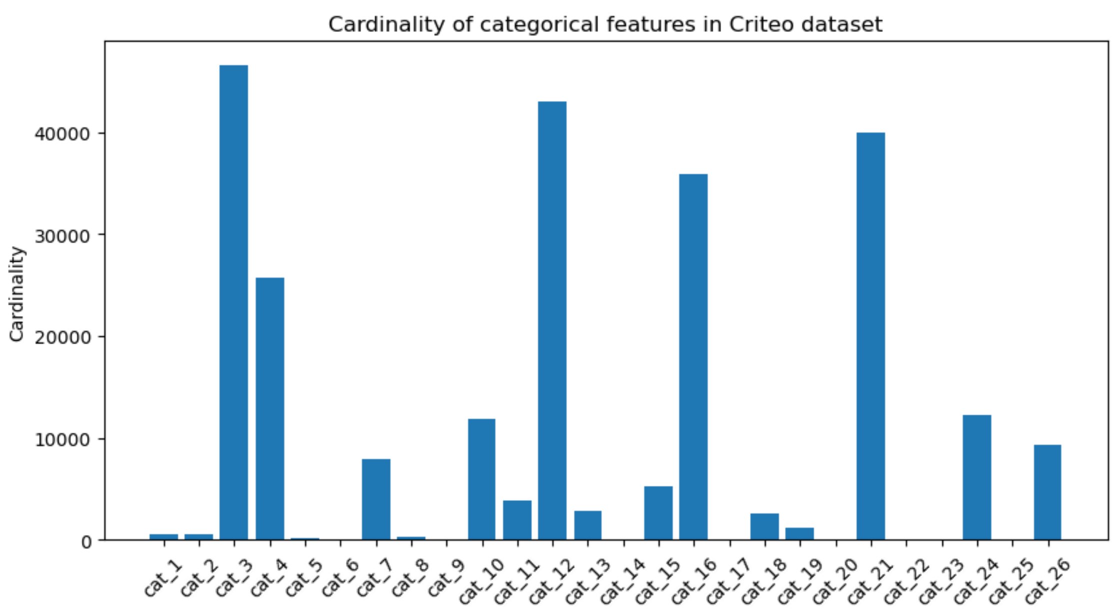
```
   cat_25_missing  cat_26_missing
0               1               1
1               1               1
2               1               1
3               1               1
4               0               0

[5 rows x 64 columns]
```

3.0.6 Sparse Multi-Value Categorical Features

As already mentioned above, ad marketplace data often contains sparse categorical features, which make signal detection extremely difficult in shallow modelling frameworks. Below I show examples from each dataset

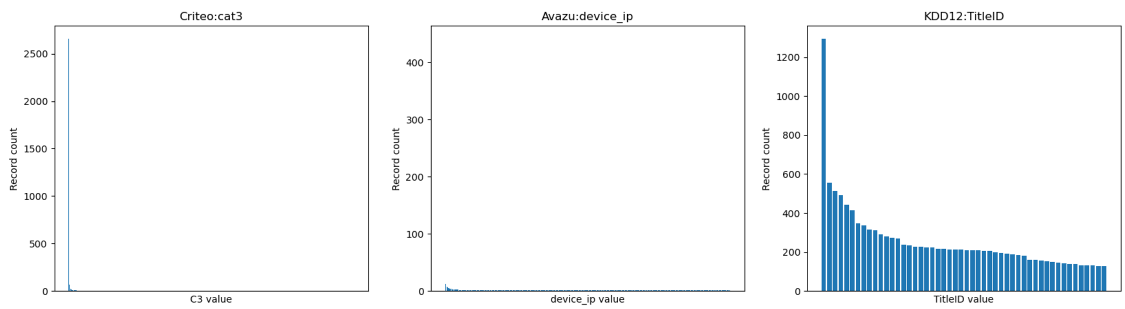







```
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\3506136802.py:2:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  criteo_cat1 = criteo.groupby('cat_3').agg({'click':'count'}).rename(columns={'
```

```
click':'count'}).sort_values('count', ascending=False).reset_index()
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\3506136802.py:3:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  avazu_site_domain = avazu.groupby('device_ip').agg({'click':'count'}).rename(c
olumns={'click':'count'}).sort_values('count', ascending=False).reset_index()
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\3506136802.py:4:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  kdd12_advertiser_id = kdd12.groupby('TitleID').agg({'Click':'count'}).rename(c
olumns={'Click':'count'}).sort_values('count',
ascending=False).reset_index().astype({'TitleID':str, 'count':int})
```

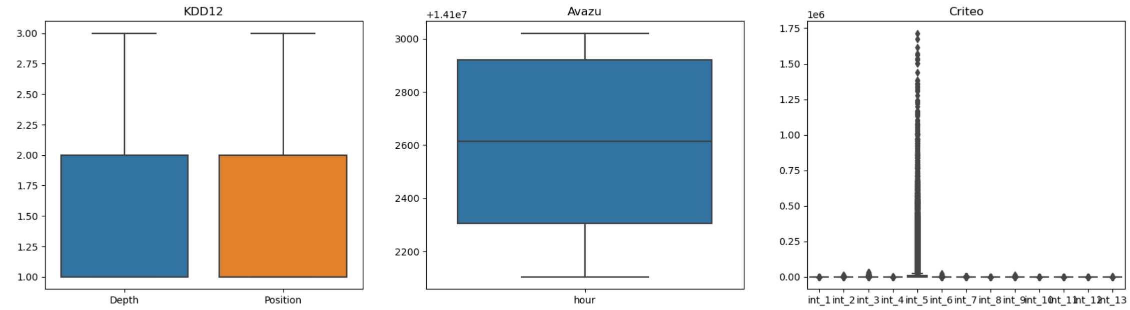


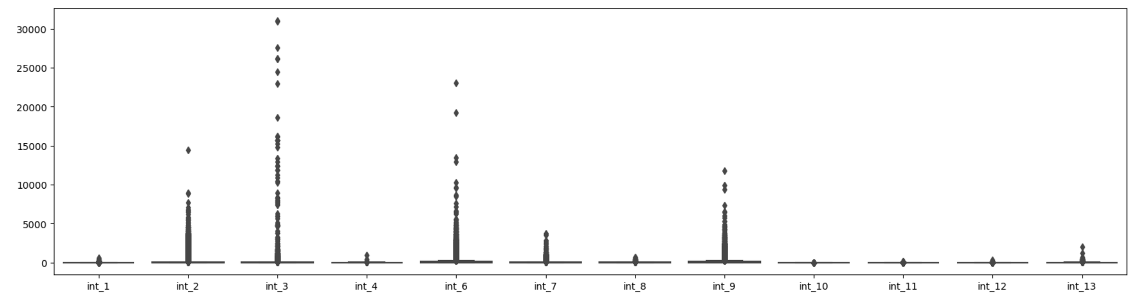
A common remidy to the above issue is to *bin* the categorical feature values before one-hot encoding or embedding, according to some given threshold (Cite Song, Others). This essentially means that for a given threshold t, we retain only the values for the multi-value categorical features that have more than t occurances in the dataset. (Cite Song) Reccomends usign, setting $t = 10, 5, 10$ for Criteo, KDD12 and Avazu respectively. However, due to computational limitations, below I proceed by limiting the *maximum number of OHE verctor dimensionality* to 20 for each dataset.

```
Before one-hot encoding:
KDD12 shape: (100000, 12)
Avazu shape: (100000, 24)
Criteo shape: (100000, 64)

After one-hot encoding:
KDD12 shape: (100000, 164)
Avazu shape: (100000, 345)
Criteo shape: (100000, 490)
```

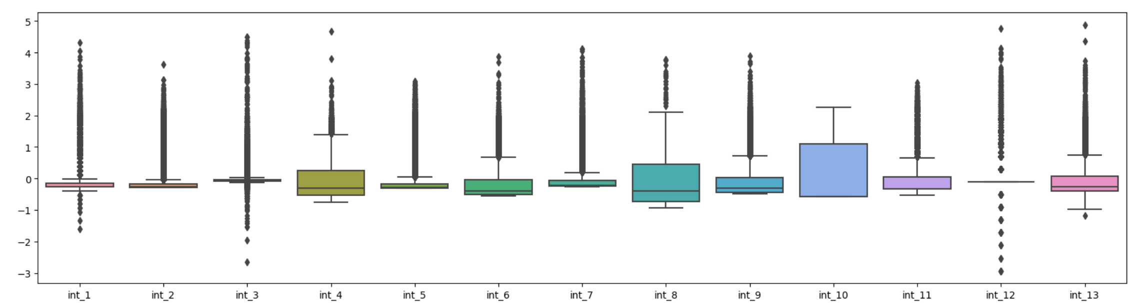
3.0.7 High Variance Numerical outliers

Below I check the distributions of the numerical features in the datasets





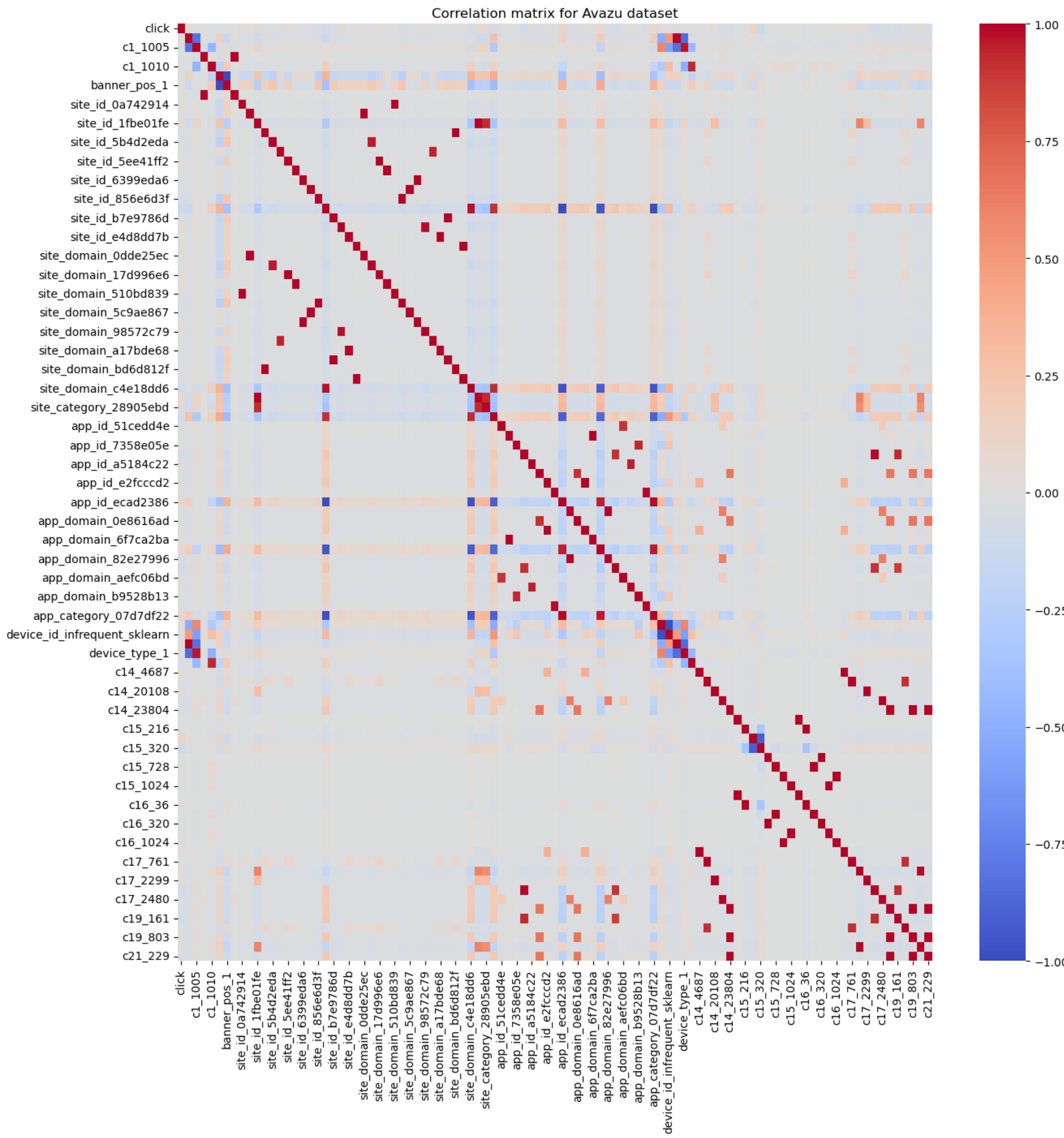
Due to the high variance of numerical features in the Criteo dataset, it is necessary to transform these variable in order to ease the training of deep NN's. As done be (Cite Song and Wang, and the winner of the Criteo Competition), we will proceed by applying the transform $\log^2(z)$ if $z > 2$, and where z is the standardized numerical value.



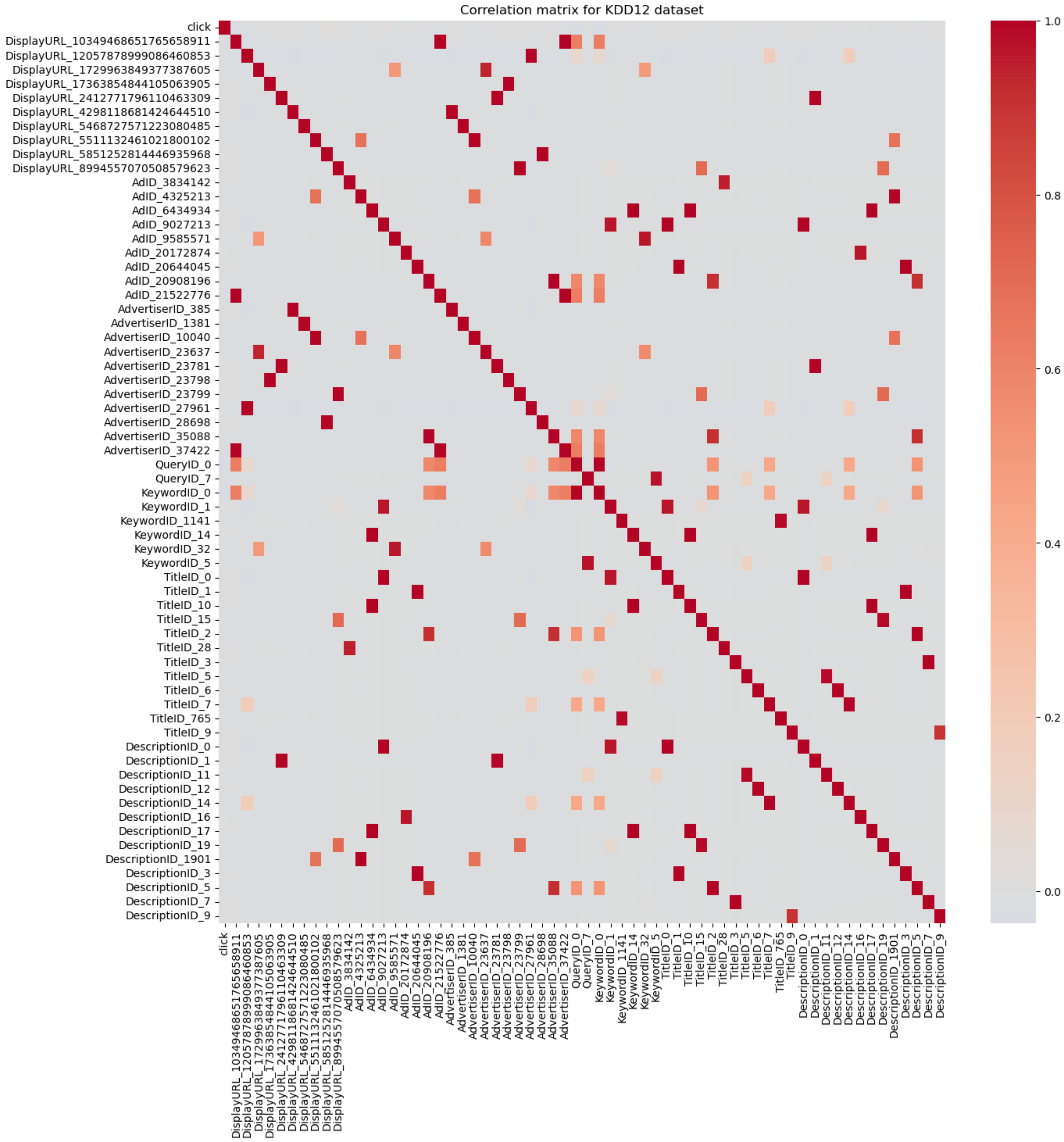
3.0.8 Correlation Analysis

Below I conduct a correlation analysis of the features to the Click-Through rate

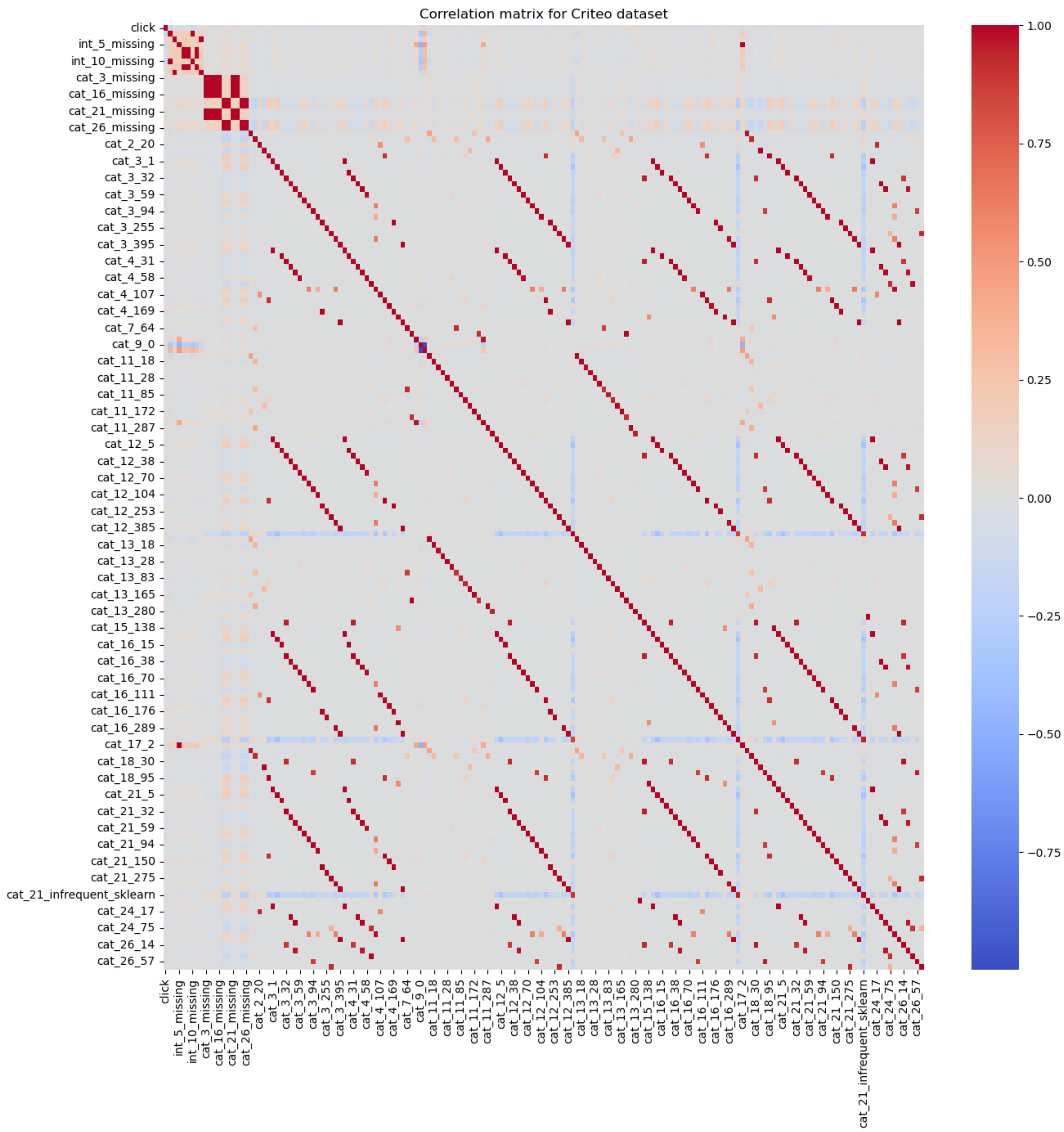
```
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\1548746507.py:6:
FutureWarning: Series.__setitem__ treating keys as positions is deprecated. In a
future version, integer keys will always be treated as labels (consistent with
DataFrame behavior). To set a value by position, use `ser.iloc[pos] = value`
  mask[0] = True # Keep the click column
```




```
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\1548746507.py:18:
FutureWarning: Series.__setitem__ treating keys as positions is deprecated. In a
future version, integer keys will always be treated as labels (consistent with
DataFrame behavior). To set a value by position, use `ser.iloc[pos] = value`
  mask[0] = True # Keep the click column
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\1548746507.py:23:
FutureWarning: Series.__setitem__ treating keys as positions is deprecated. In a
future version, integer keys will always be treated as labels (consistent with
DataFrame behavior). To set a value by position, use `ser.iloc[pos] = value`
  mask[0] = True # Keep the click column
```



Correlation matrix for KDD12 dataset
click
DisplayURL_10349468651765658911
DisplayURL_12057878999086460853
DisplayURL_1729963849377387605
DisplayURL_17363854844105063905
DisplayURL_2412771796110463309
DisplayURL_4298118681424644510
DisplayURL_5468727571223080485
DisplayURL_5511132461021800102
DisplayURL_5851252814446935968
DisplayURL_8994557070508579623
AdID_3834142
AdID_4325213
AdID_6434934
AdID_9027213
AdID_9585571
AdID_20172874
AdID_20644045
AdID_20908196
AdID_21522776
AdvertiserID_385
AdvertiserID_1381
AdvertiserID_10040
AdvertiserID_23637
AdvertiserID_23781
AdvertiserID_23798
AdvertiserID_23799
AdvertiserID_27961
AdvertiserID_28698
AdvertiserID_35088
AdvertiserID_37422
QueryID_0
QueryID_7
KeywordID_0
KeywordID_1
KeywordID_1141
KeywordID_14
KeywordID_32
KeywordID_5
TitleID_0
TitleID_1
TitleID_10
TitleID_15
TitleID_2
TitleID_28
TitleID_3
TitleID_5
TitleID_6
TitleID_7
TitleID_765
TitleID_9
DescriptionID_0
DescriptionID_1
DescriptionID_11
DescriptionID_12
DescriptionID_14
DescriptionID_16
DescriptionID_17
DescriptionID_19
DescriptionID_1901
DescriptionID_3
DescriptionID_5
DescriptionID_7
DescriptionID_9
1.0
0.8
0.6
0.4
0.2
0.0



Some very high correlations between some of the features across fields in all three datasets. This possibly points to there being potential for dimensionality reduction across this feature set.

Unfortunately, from the correlation heatmaps, there appears to be little to no correlation between the first-order features and the target click variable.

4 Modelling

In this section I will compare the perfomance of two shallow modelling approaches (Logistic Regression and Factorization Machines) to a naive DNN for CTR prediction. As with (Cite Song and Wang), I will use the **Area Under the ROC Curve** and **Logloss** measures to compare the performance of the different modelling approaches on the test set.

```
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\1507951586.py:3: DtypeWarning:
Columns (7,10) have mixed types. Specify dtype option on import or set
low_memory=False.
  avazu_standardized = pd.read_csv('./data/avazu/avazu_normed_labels.csv')
C:\Users\marti\AppData\Local\Temp\ipykernel_14764\1507951586.py:4: DtypeWarning:
Columns (27) have mixed types. Specify dtype option on import or set
low_memory=False.
  criteo_standardized = pd.read_csv('./data/criteo/criteo_normed_labels.csv')
```

5 Simulatinons

6 Summary of findings

7 Suggested Future Research

8 References

- eMarketer. (2023). Digital advertising spending worldwide from 2021 to 2027 (in billion U.S. dollars) . Statista. Statista Inc.. Accessed: June 09, 2024. https://www-statista-com.iclibezp1.cc.ic.ac.uk/statistics/237974/online-advertising-spending-worldwide/
- Aden, Yi Wang. (2012). KDD Cup 2012, Track 2. Kaggle. https://kaggle.com/competitions/kddcup2012-track2
- Steve Wang, Will Cukierski. (2014). Click-Through Rate Prediction. Kaggle. https://kaggle.com/competitions/avazu-ctr-prediction
- Jean-Baptiste Tien, joycenv, Olivier Chapelle. (2014). Display Advertising Challenge. Kaggle. https://kaggle.com/competitions/criteo-display-ad-challenge