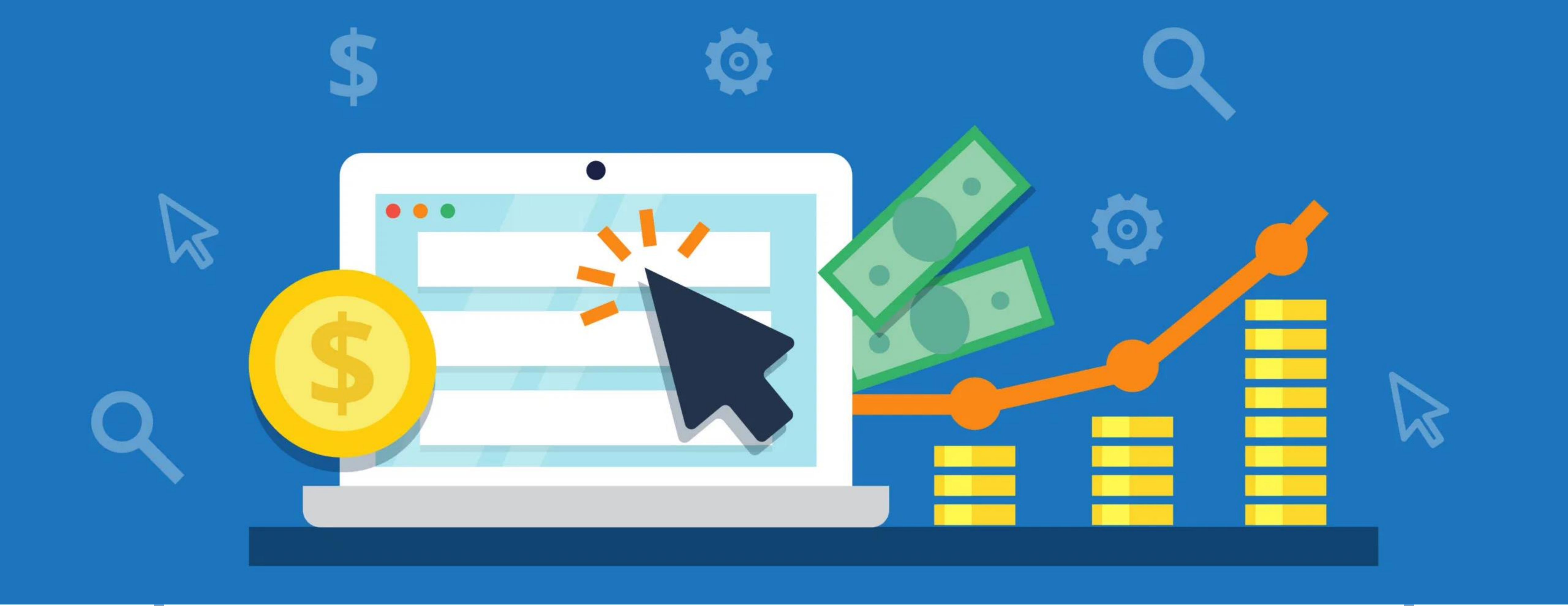


Ad Clicks

A Machine Learning Project

by João Gil Ribeiro – 32399, Lina Francisca Araújo – 31892, Sebastião Spínola – 44352, Tomás Borges - 32075

The Business Problem

Digital advertisement is now a main strategy for companies, going beyond banner ads with basic targeting.

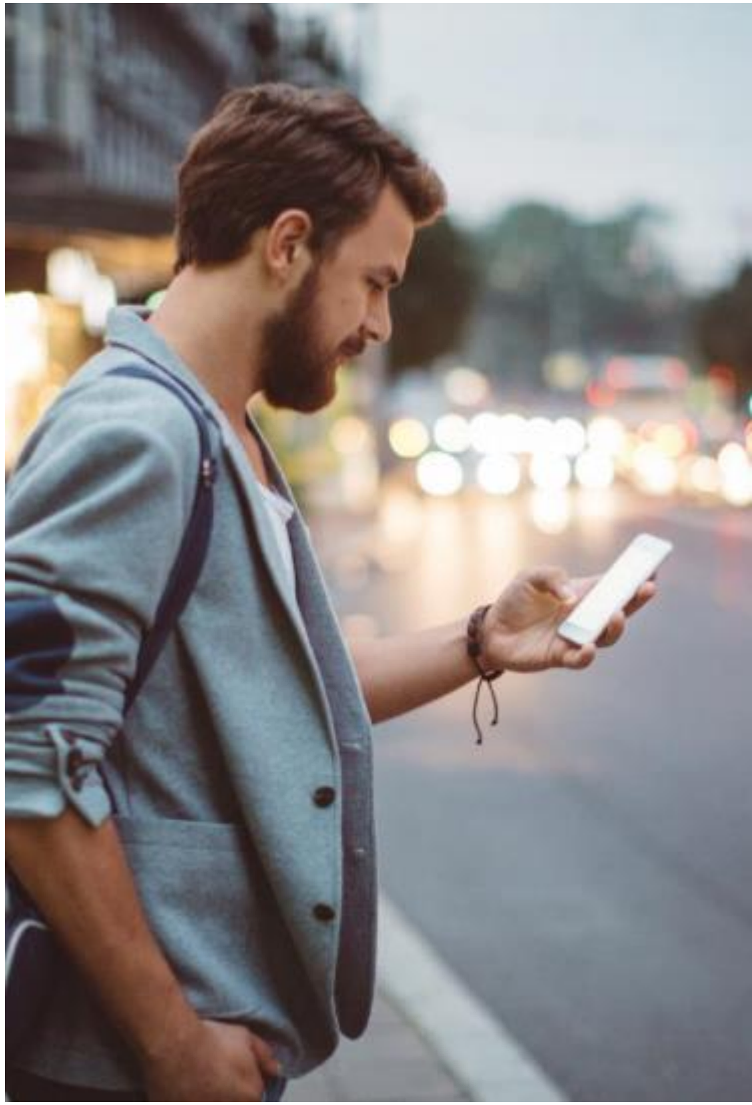
Using a dataset containing ads shown during 10 days by a large service provider of online advertising and digital marketing.

We want to predict if a user will **click** or **not click** on an ad with special attention to the role of the website, the position and display of the add and other features that can be controlled by the advertiser.

The Machine Learning Problem

Due to the characteristics of the data this is considered an **unbalanced supervised classification** problem.

Target Variable:
83% (82 962) – not click
17% (17 031) - click



Features

The dataset **24 features by 100.000 instances** can be divided into **7 categories**:

1. User identifier:
- *id*: ad unique identifier

2. Target Variable:
- *click*: 0/1 for non-click / click

3. Website Features:
- *banner_pos*: ad's banner position on website
- *site_id*: website unique identifier
- *site_domain*: website link
- site_category: group of websites with similar content

4. App Features:
- *app_id*: app unique identifier
- *app_domain*: app address / name
- app_category: group of apps with similar content

5. Device Features:
- *device_id*: device unique identifier
- *device_ip*: another device unique identifier
- device_model: refers to the manufactures (Apple, Samsung, etc..) device models such as iPhone 8, iPhone X, Galaxy 8
- *device_type*: label to match device to series or model
- *device_conn_type*: label associated to the type of device connection.

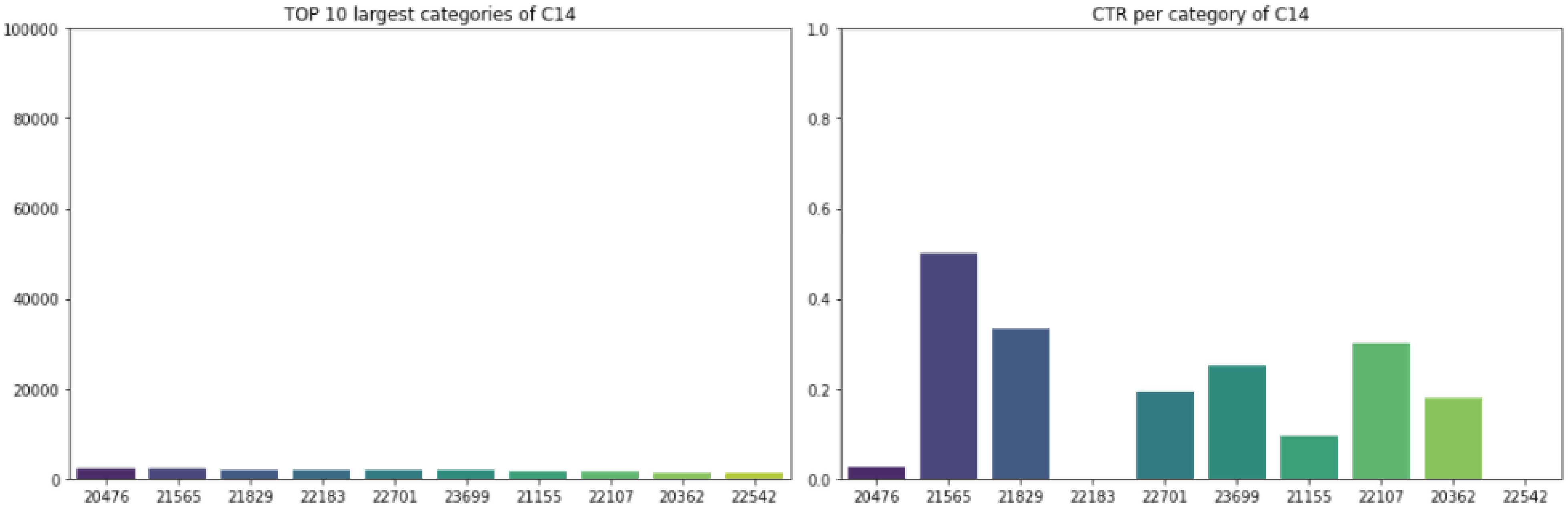
6. Hour / Time Features:
- *hour*: format is YYMMDDHH

7. Anonymised Categorical Features:
- *C1*
- C14 to C21

Exploratory Data Analysis

Main Steps:

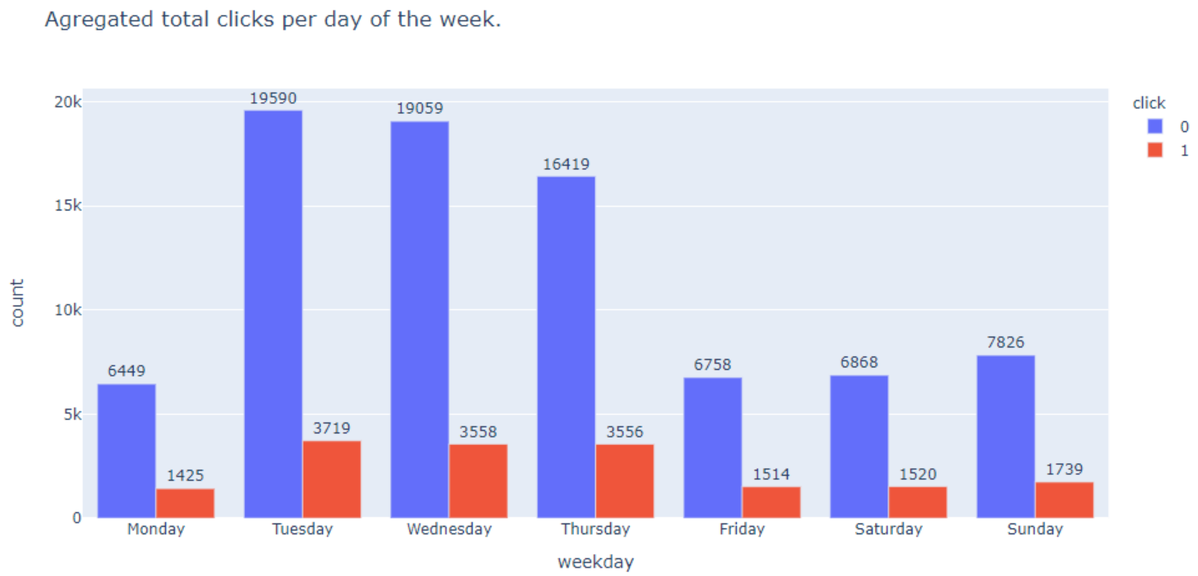
- Breakdown of clicks by time (hour and day)
- Click-through rate as the core metric of our problem to understand relationships between features and the target -> *How many people who've seen your ad end up clicking on it*
- Understanding the meaning of variables (Some variables are anonymous)
- Understand the relevance of the values inside each anonymous categorical feature

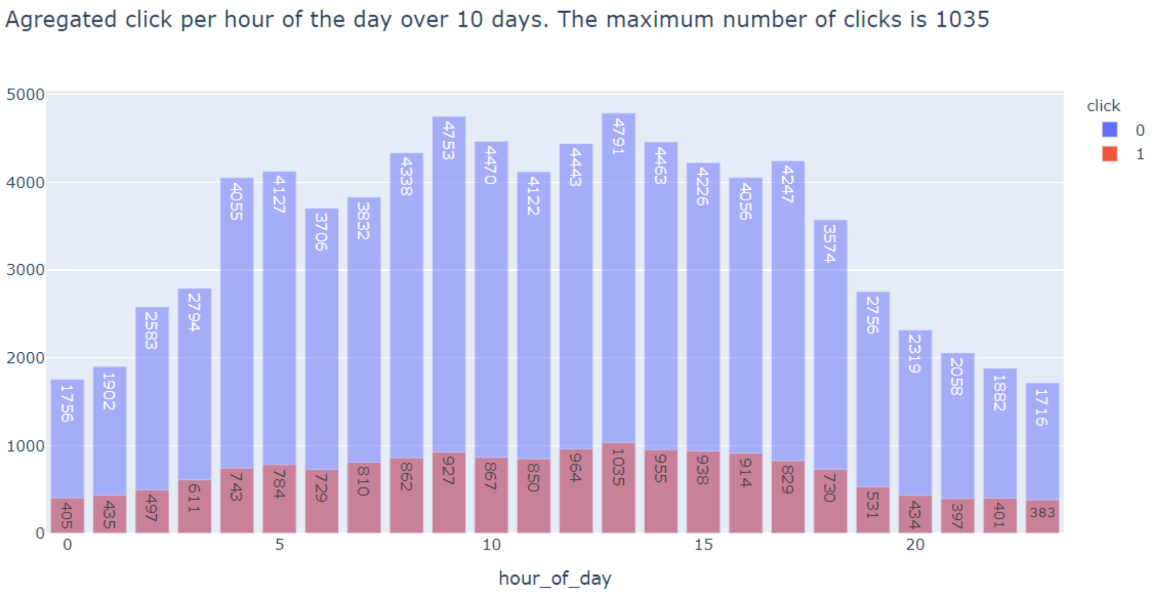


Feature Selection

Dropped Features

- **Id** - unique index, it has 100.000 unique observations, same as total rows of the dataset
- **Device Id** – a unique string for the user's device
- **Device Ip** – unique address to identify the device, not constant across public and individual networks
- **Hour** – DataTime feature broken down into **hour_of_day**, **weekday** and **day**
- **Day** – day of the week in our 10 days sample period (too short to take conclusions and possibly not generalizable). By using ***hour_of_day*** **and** ***weekday*** **we get an average over the days and thus make this possible effect less significant**





Feature Engineering

Dataset Shape after Feature Selection: (100000, 22)

Transformations applied:

- **One Hot Encoding** of: ['device_type', 'device_conn_type', 'C18’]
- **Label Encoding** of: ['site_category', 'app_domain', 'app_category’, 'weekday', 'C1', 'banner_pos', 'C15', 'C16', 'C19', 'C21’]
- **Target Encoding** of: ['site_id', 'site_domain', 'app_id', 'device_model', 'C14', 'C17', 'C20’]

Dataset Shape After Encoding: (100000, 49)

```
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 49 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   click               100000 non-null  int64
 1   site_id             100000 non-null  int32
 2   site_domain         100000 non-null  int32
 3   site_category       100000 non-null  int32
 4   app_id              100000 non-null  int32
 5   app_domain          100000 non-null  int32
 6   app_category        100000 non-null  int32
 7   device_model        100000 non-null  int32
 8   C14                 100000 non-null  int64
 9   C15                 100000 non-null  int64
 10  C16                 100000 non-null  int64
 11  C17                 100000 non-null  int64
 12  C19                 100000 non-null  int64
 13  C20                 100000 non-null  int64
 14  C21                 100000 non-null  int64
 15  hour_of_day         100000 non-null  int64
 16  weekday_0           100000 non-null  uint8
 17  weekday_1           100000 non-null  uint8
 18  weekday_2           100000 non-null  uint8
 19  weekday_3           100000 non-null  uint8
 20  weekday_4           100000 non-null  uint8
 21  weekday_5           100000 non-null  uint8
 22  weekday_6           100000 non-null  uint8
 23  C1_1001             100000 non-null  uint8
 24  C1_1002             100000 non-null  uint8
 25  C1_1005             100000 non-null  uint8
 26  C1_1007             100000 non-null  uint8
 27  C1_1008             100000 non-null  uint8
 28  C1_1010             100000 non-null  uint8
 29  C1_1012             100000 non-null  uint8
 30  banner_pos_0        100000 non-null  uint8
 31  banner_pos_1        100000 non-null  uint8
 32  banner_pos_2        100000 non-null  uint8
 33  banner_pos_3        100000 non-null  uint8
 34  banner_pos_4        100000 non-null  uint8
 35  banner_pos_5        100000 non-null  uint8
 36  banner_pos_7        100000 non-null  uint8
 37  device_type_0       100000 non-null  uint8
 38  device_type_1       100000 non-null  uint8
 39  device_type_4       100000 non-null  uint8
 40  device_type_5       100000 non-null  uint8
 41  device_conn_type_0  100000 non-null  uint8
 42  device_conn_type_2  100000 non-null  uint8
 43  device_conn_type_3  100000 non-null  uint8
 44  device_conn_type_5  100000 non-null  uint8
 45  C18_0               100000 non-null  uint8
 46  C18_1               100000 non-null  uint8
 47  C18_2               100000 non-null  uint8
 48  C18_3               100000 non-null  uint8
dtypes: int32(7), int64(9), uint8(33)
```

Modeling

1. Evaluation Metric
2. Modeling Pipeline
3. Four Models
4. Choosing the Best Model
5. Feature Importance
6. Test Best Model

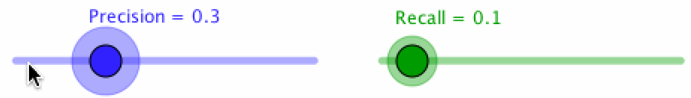
Evaluation Metrics

We chose to **maximize the f1-score given it is the harmonic mean between precision and recall.**

This **balance** is important given the context of our **Business Problem which relies on good recall and precision of the "click"** target as it is our interest to be able to predict if a given user will click or not on the ad.

AUC is also a widely used metric to compare binary classification models.

- AUC is **not as good a measure for Imbalanced Datasets**
- We can have a model with a high AUC but recalling very few True Positives.
- The AUC is high only because there are very few predictions for the True Positive Class, and these are mostly correct.



$$F1 = \frac{2 * precision * recall}{precision + recall}$$

$$F1 = \frac{2 \times 0.3 \times 0.1}{0.3 + 0.1} \quad \therefore F1 = 0.15$$

Harmonic mean is conservative mean compared to Arithmetic meand and geometric mean.
It means that Harmonic mean is nearest to the smallest of the input numbers.

Modeling Pipeline

Given our **dataset is unbalanced**, and **after preliminar modeling showed that without rebalancing the model performance was severely hindered with an over importance of the majority class.**

Target Encoding is an encoding method to **reduce the effect of high cardinality features**. Which were pre-selected as being features with **more than 150 unique values** and append to **target_enc** list

`Pipeline`

> *'sampling'* → `RandomOverSampler()` , `RandomOverSampler()` , `SMOTE()`
> *'transformer'* → `TargetEncoder(cols=target_enc)`
> *'scaler'* → `StandardScaler()`
> *'classifier'* → `KNeighborsClassifier()` , `RandomForestClassifier()` , `CatBoostClassifier()`

KNN Classifier

- **Simple** supervised classification algorithm
- No assumptions on the data distribution, hence it is **non-parametric**
- It keeps all the training data to make future predictions
- Computes the similarity between an input sample and each training instance.

CatBoost

- **Optimized for categorical features**
- GPU training
- Uses bagged and smoothed version of target encoding for categorical variables

Random Forest

- Generally, it is a **fast**, **simple** and **flexible** algorithm
- Has a high classification accuracy (hard to build a bad model)
- Gives information about feature importance
- Reduces model variance

Stacking Classifier

- Combines multiple classification models via a meta-classifier, learning its strengths and weaknesses and **delivers the best outcome**
- Models used:
  - Best from KNN
  - Best from Random Forest
  - Best from CatBoost
- Voting Classifier used to select best one.

Choosing the Best Model

Grid Search through the Best Cross-Validation models:

- KNN Classifier
- Random Forest Classifier's
- CatBoost
- Stacking (of all the above)

Pipeline

```
('sampling', RandomUnderSampler(random_state=42)),
('transformer',
 TargetEncoder(cols=['site_id', 'site_domain', 'app_id',
                     'device_model', 'C14', 'C17', 'C20'])),
('scaler', StandardScaler()),
('classifier',
 RandomForestClassifier(class_weight='balanced', max_depth=58,
                        min_samples_leaf=37,
                        min_samples_split=13, n_estimators=750,
                        random_state=42))])
```

	param_classifier	mean_test_f1_score	std_test_f1_score	rank_test_f1_score
3	RandomForestClassifier(class_weight='balanced'...	0.396958	0.005089	1
1	RandomForestClassifier(max_depth=39, min_sampl...	0.396706	0.003284	2
2	RandomForestClassifier(max_depth=27, min_sampl...	0.396676	0.005481	3
5	VotingClassifier(estimators=[('knn_optimized',...	0.393913	0.004823	4
4	<catboost.core.CatBoostClassifier object at 0x...	0.393792	0.004650	5
0	KNeighborsClassifier(n_neighbors=43)	0.385879	0.004114	6

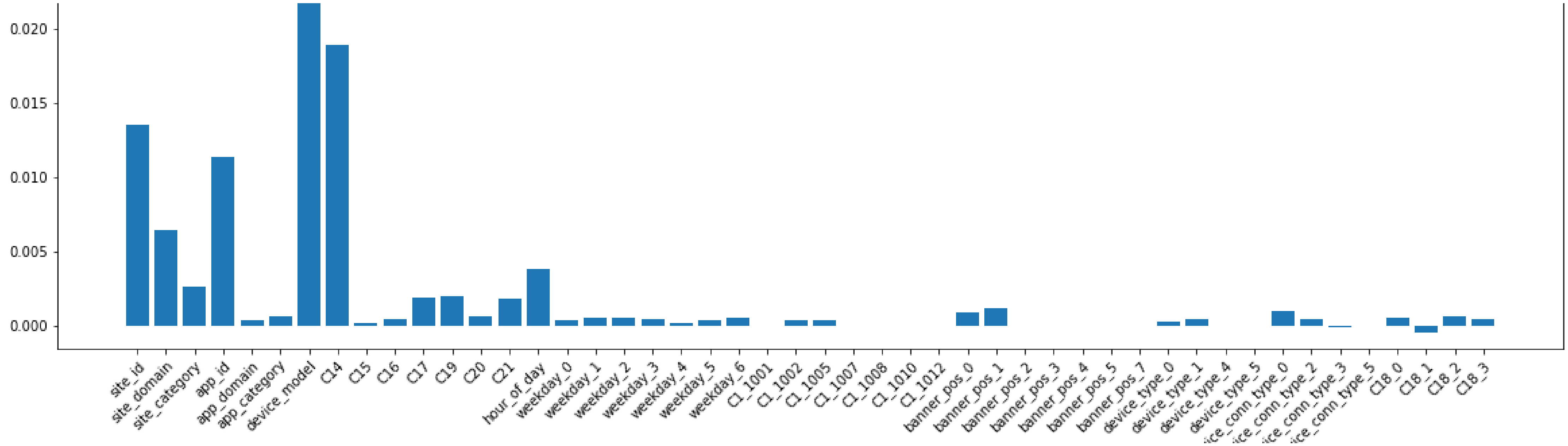
Feature Importance

Permutation

- Scoring f1 to correct the negative impacts
- Feature importance from the train data
- Clearly, some features are not relevant

Most Important Features

- The most important features are the most interpretable ones, **except C14**
- **Device Model, App ID, Site ID, Device Domain.**
- **Hour of day** has some importance.



Test Best Model

Main Target: Optimize F1 score

F1 Score = 0,40

- Initial models – simple decision trees – were at 0.27

Recall = 0,73

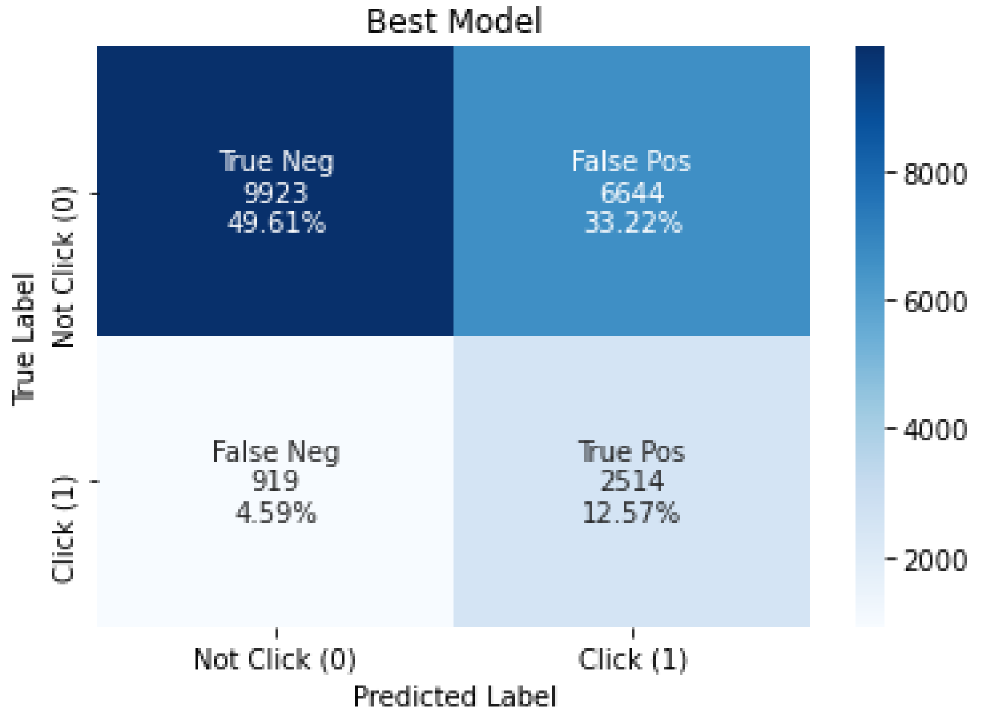
- Able to minimize the False Negatives

Precision = 0,27

- Proved difficult to increase, led to a high number of False Positives

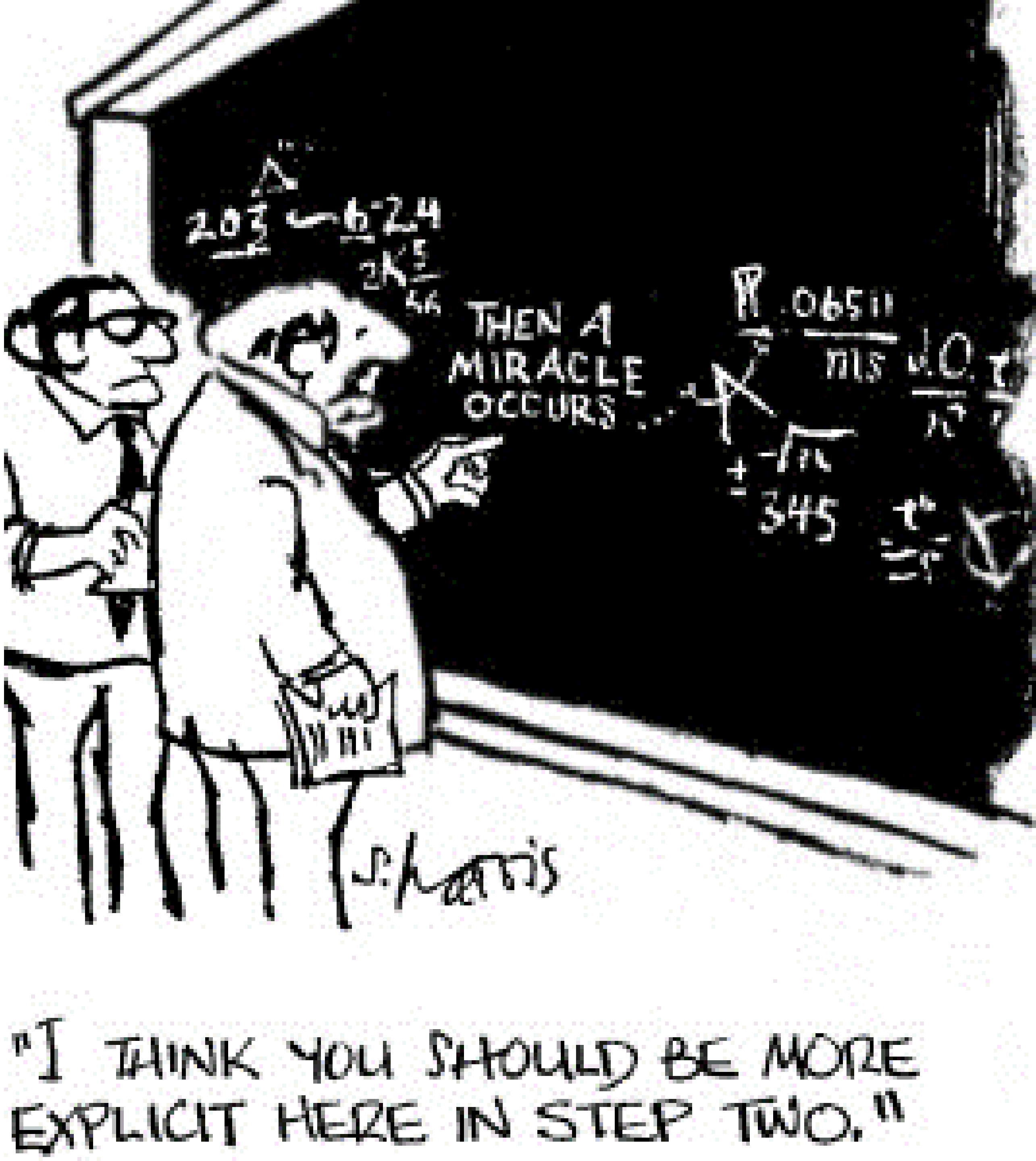
AUC = 0,72

	precision	recall	f1-score	support
0	0.92	0.60	0.72	16567
1	0.27	0.73	0.40	3433
accuracy			0.62	20000
macro avg	0.59	0.67	0.56	20000
weighted avg	0.81	0.62	0.67	20000



Interpretability

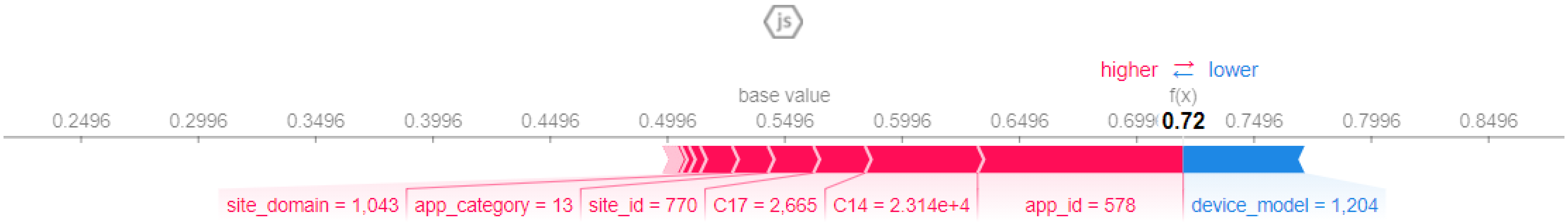
1. SHAP Single Obs.
2. Feature Importance
3. Summary Plot
4. Partial Dependance Plot (PDP)



Model Interpretation – SHAP Single Obs.

The output prediction is 1, which means **the model classifies this observation** as a **`click`.** The base value is 0.5496, **Feature values that push towards a `no click` are in blue** - `device_model`.

Feature values **increasing the prediction are in pink, namely `app_id` and `C14`.**



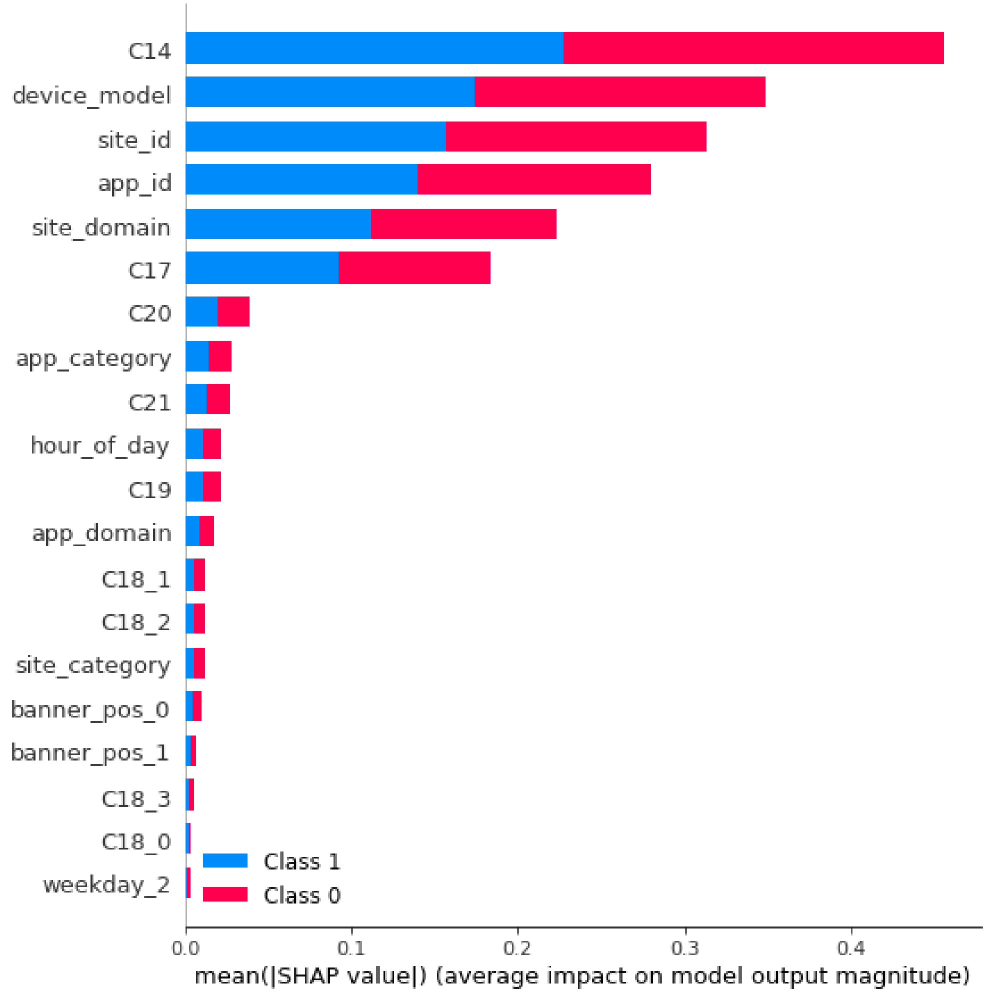
Model Interpretation – Feature Importance

SHAP feature importance measured as the mean absolute Shapley values.

The **C14 anonymous category** was the most important feature, **changing the predicted absolute click probability on average by 45 percentage points** (0.45 on x-axis).

Followed by **device_model** at 35, **site_id** at 31, **app_id** at 29, **site_domain** 25 and C17 at 19 percentage points

NOTE: Permutation feature importance is based on the decrease in model performance. SHAP is based on magnitude of feature attributions.

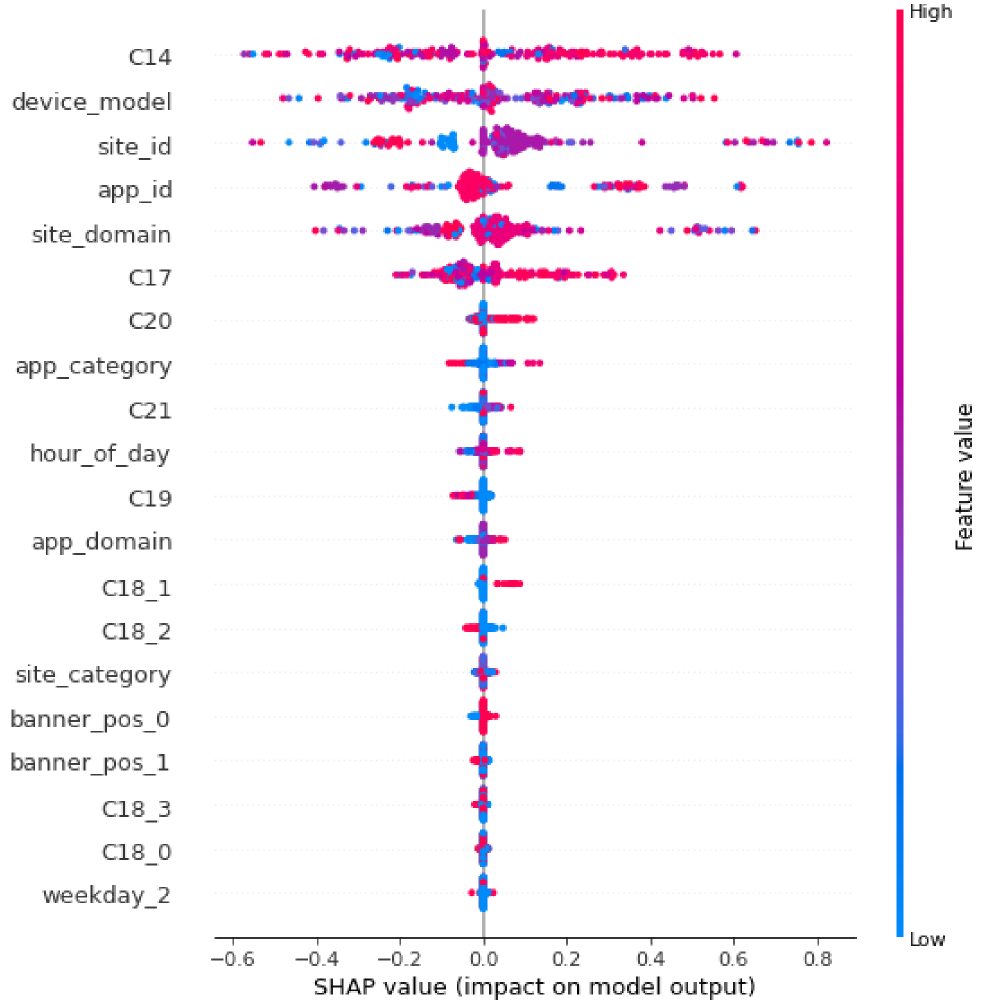


Model Interpretation – Summary Plot

Positive SHAP value impact on the model output leads to "**pushing" to 1 the probability of the customer having clicked click = 1** and a negative SHAP value impact leads to the opposite push.

We can see that **C14, device_model, site_id, app_id, site_domain, C17 and C20** are the features with the **highest impact** on the model.

site_id has that the **middle feature values affect the Shap values positively,** between 0.0 and 0.2 thus pushing click probability slightly, while high or low feature values tend to be on the extremes.



Model Interpretation – PDP

We chose the two best features that were interpretable (excluding anonymized features).

- **device model** and **site id**
- This plot helps to map users who user are more likely to click - **lighter regions –** or not to click - **darker regions -** based on the interaction between the two features.

For **site_id** at 0.558 and values of **device_model around 0.5** - 0.51, 0.5, 0.491 and 0.494 – there's a **higher probability of the outcome being click.**