Get the data

In [ ]:
```
!gdown --id 1HVSazFk8m553VWPjFnZZ-YfJA_KecPea
!unzip translated_data_updated.zip
```

```
Downloading...
From: https://drive.google.com/uc?id=1HVSazFk8m553VWPjFnZZ-YfJA_KecPea
To: /content/translated_data_updated.zip
100% 122M/122M [00:01<00:00, 105MB/s]
Archive:  translated_data_updated.zip
   creating: data_translated/
  inflating: data_translated/coupon_visit_train.csv
  inflating: data_translated/coupon_list_train.csv
  inflating: data_translated/prefecture_locations.csv
  inflating: data_translated/coupon_area_test.csv
  inflating: data_translated/coupon_detail_train.csv
  inflating: data_translated/coupon_area_train.csv
  inflating: data_translated/user_list.csv
  inflating: data_translated/coupon_list_test.csv
```

In [ ]:
```
%%capture
!pip install tensorflow_decision_forests
```

In [ ]:
```
# imports
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

import tensorflow as tf
import tensorflow_decision_forests as tfdf
from tensorflow import keras
import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

sns.set_theme(context='notebook', style='darkgrid')
mpl.rcParams['figure.figsize'] = (12, 10)
colors = plt.rcParams['axes.prop_cycle'].by_key()['color']
```

```
WARNING:root:TF Parameter Server distributed training not available.
```

In [ ]:
```
# Important Note:
# Visits = browsing history in the training period. No test set available.
# Purchases = purchase history in the training period. No test set available.

df_users        = pd.read_csv('data_translated/user_list.csv')
df_c_list_train = pd.read_csv('data_translated/coupon_list_train.csv')
df_c_list_test  = pd.read_csv('data_translated/coupon_list_test.csv')
df_area_train   = pd.read_csv('data_translated/coupon_area_train.csv')
df_area_test    = pd.read_csv('data_translated/coupon_area_test.csv')
df_visit_train  = pd.read_csv('data_translated/coupon_visit_train.csv')
df_purch_train  = pd.read_csv('data_translated/coupon_detail_train.csv')
df_locations    = pd.read_csv('data_translated/prefecture_locations.csv')
```

Feature Engineering

Since TF Decision Forests can handle categorical variables just fine, we're not doing much preprocessing.

In [ ]:
```
# rename SEX_ID column, change to categorical value (0 Male, 1 Female)
df_users['SEX'] = df_users['SEX_ID'].replace('f', 1)
df_users['SEX'] = df_users['SEX'].replace('m', 0)
```

In [ ]:
```
# create a categorical variable for age group:
# 14-21, 22-35, 36-49, 50-65, 66-75, 76-90
def age_cat(age):
  if age <= 21:
    return 0
  elif age <= 35:
    return 1
  elif age <= 49:
    return 2
  elif age <= 65:
    return 3
  elif age <= 75:
    return 4
  elif age <= 90:
    return 5
  else:
    return 6

lbl_age_ranges = ['14-21', '22-35', '36-49', '50-65', '66-75', '76-90']

df_users['AGE_GROUP'] = [age_cat(a) for a in df_users['AGE']]
```

In [ ]:
```
# Model Input Features
# For each user who purchased a coupon...

# Gender, Age, Prefecture, Coupon Genre, Coupon Prefecture, Price Rate, Catalog

##############################
# BUILD DF_TRAIN DATAFRAME  #
##############################
df_visit_train = df_visit_train.rename(columns={'VIEW_COUPON_ID_hash': 'COUPON_I
df_train = df_visit_train.join(df_users.set_index('USER_ID_hash'), on='USER_ID_h
df_train = df_train.join(df_c_list_train.set_index('COUPON_ID_hash'), on='COUPON
# get a subset of the training columns and rename them
df_train = df_train[['AGE_GROUP', 'SEX', 'PREF_NAME_EN', 'KEN_NAME_EN', 'GENRE_N
df_train.columns = ['age_group', 'sex', 'user_prefecture', 'coupon_prefecture',
# NaN preprocessing
```

In [ ]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2833180 entries, 0 to 2833179
Data columns (total 9 columns):
 #   Column             Dtype
```

```
---  ------              -----
 0   age_group           int64
 1   sex                 int64
 2   user_prefecture     object
 3   coupon_prefecture   object
 4   genre               object
 5   capsule             object
 6   discount_rate       float64
 7   discount_price      float64
 8   purchased           int64
dtypes: float64(2), int64(3), object(4)
memory usage: 194.5+ MB
```

In [ ]:
```
# Train the model!
#df_train_set, df_test_set = train_test_split(df_train, test_size=0.2, stratify=

ds_train_set = tfdf.keras.pd_dataframe_to_tf_dataset(df_train, label='purchased'
model = tfdf.keras.GradientBoostedTreesModel(num_trees=500,
                                             growing_strategy='BEST_FIRST_GLOBAL
                                             max_depth=8, split_axis='SPARSE_OBL

model.fit(ds_train_set)
```

```
44269/44269 [==============================] - 83s 2ms/step
```

Out[ ]:
```
<keras.callbacks.History at 0x7f6d96b4c490>
```

In [ ]:
```
# START HERE - run cells 108-114
tfdf.model_plotter.plot_model_in_colab(model, tree_idx=0)
```

Out[ ]:

Get User's Purchased Coupons

In [ ]:
```
# preprocess the test set to make it a little faster
test_coupons = df_c_list_test
test_coupons = test_coupons[['PRICE_RATE', 'DISCOUNT_PRICE', 'COUPON_ID_hash', '
coupon_ids = test_coupons['COUPON_ID_hash']
def merge_user_with_test_coupons(user):
  df = pd.DataFrame()

  df['user_id'] = user['USER_ID_hash']
  df['coupon_id'] = test_coupons['COUPON_ID_hash']
  df['age_group'] = user['AGE_GROUP']
  df['sex'] = user['SEX']
  df['user_prefecture'] = np.array(user['PREF_NAME_EN']).astype(np.object)
  df['coupon_prefecture'] = test_coupons['KEN_NAME_EN']
  df['genre'] = test_coupons['GENRE_NAME_EN']
  df['capsule'] = test_coupons['CAPSULE_TEXT_EN']
  df['discount_rate'] = test_coupons['PRICE_RATE']
  df['discount_price'] = test_coupons['DISCOUNT_PRICE']

  df['sex'] = df['sex'].replace('m', 0)
  df['sex'] = df['sex'].replace('f', 1)

  return df
```

In [ ]:
```
from tqdm import tqdm
all_predictions = []
```

```
for i, u in tqdm(df_users.iterrows(), total=len(df_users)):
  user_coupons = merge_user_with_test_coupons(u)
  ds_user_coupons = tfdf.keras.pd_dataframe_to_tf_dataset(user_coupons.drop(colu
  preds = model.predict(ds_user_coupons)
  preds = preds.ravel()

  df_pred = pd.DataFrame(data={'coupon_id': coupon_ids, 'likelihood': preds}, co
  top_coupons = df_pred.sort_values(by='likelihood', ascending=False)[:10]

  coupon_string = ' '.join(top_coupons['coupon_id']).strip()
  all_predictions.append({'USER_ID_hash': u['USER_ID_hash'], 'PURCHASED_COUPONS'

submission_df = pd.DataFrame.from_dict(all_predictions)
submission_df.to_csv('submission_decision_tree.csv', header=True, index=False)

submission_df
```

```
100%|██████████| 22873/22873 [45:53<00:00,  8.31it/s]
```

Out[ ]:

	USER_ID_hash	PURCHASED_COUPONS
0	d9dca3cb44bab12ba313eaa681f663eb	5e47b887e154f746883013f863c3ffe1 27741884a086e...
1	560574a339f1b25e57b0221e486907ed	5e47b887e154f746883013f863c3ffe1 27741884a086e...
2	e66ae91b978b3229f8fd858c80615b73	87ffb19277d6ca4065a492508af1ae27 5e47b887e154f...
3	43fc18f32eafb05713ec02935e2c2825	5e47b887e154f746883013f863c3ffe1 46da51ba6dd20...
4	dc6df8aa860f8db0d710ce9d4839840f	5e47b887e154f746883013f863c3ffe1 bf339b53786a8...
...	...	...
22868	2f0a2f36a9f63b6ba2fa3a7e53bef906	5e47b887e154f746883013f863c3ffe1 27741884a086e...
22869	6ae7811a9c7c58546d6a1567ab098c21	a4dbd920d68de951482b661f8d3717eb 87ffb19277d6c...
22870	a417308c6a79ae0d86976401ec2e3b04	5e47b887e154f746883013f863c3ffe1 27741884a086e...
22871	4937ec1c86e71d901c4ccc0357cff0b1	27741884a086e2864936d7ef680becc2 3d5c0b4c9e353...
22872	280f0cedda5c4b171ee6245889659571	5e47b887e154f746883013f863c3ffe1 92eb7b05f6e83...

22873 rows × 2 columns

In [ ]:
```
submission_df.to_csv('submission_gradient_boosted_hp.csv', header=True, index=Fa
```

In [ ]:
```
```